2002年5月1日発行（毎月1回1日発行）第21巻第5号 昭和57年12月6日第三種郵便物認可

マイコン

わかる！動かせる！
プログラムが組める雑誌

MAY. 2002

5

# BASIC Magazine

定価
550Yen

掲載プログラムがダウンロードできる! http://www.basicmagazine.net/

第2弾 文字・文章—いろいろな書き方・見せ方

大好評連載 見やすい楽しい誰でも使えるサイトを実現！

## 一歩先ゆく HP作り

ホームページ

編集部の学校訪問記

神奈川総合高校
進路や適正に合わせた時間割で、自覚を持った学習を可能にする

横浜市立本郷中学校
独自のテキストを駆使しながら情報教育にがんばる先生と生徒

BM基礎講座
ベーシックプログラミング教室
第2回 分岐処理と繰り返しを学ぶ

CPUの歴史と未来をさぐる！
モバイル・プロセッサがユビキタスな世界を作る！

PCハードウエアテクニック
旧型AT機を完全復活！
マザーボード換装テクニック

これからはじめるブロードバンド活用法
eAccessでのADSL設定方法

特選投稿プログラム・コーナー

- Visual Basic V6.0 攻めてくる数字軍団
- Visual Basic V6.0 ナンバープレイっす
- Visual Basic V6.0+DirectX LaserShootⅣ
- Visual Basic V6.0+DirectX a-z
- VBScript+DirectX キューブパズル
- Excel(VBA) パニックスライム移植版
- F-BASIC V6.0 感撃！イロイロ棒
- HSP V2.55 Defense-line
- Javaアプレット vegetable3
- iアプリ Pair Arrow
- PIECE Nazo(ナゾ)

AKI-PIC877モジュール用ベースボード・キットでPIC-BASIC

# アルファのブロードバンド
# フレッツ・ADSL8Mサービス開始！

**フレッツ・ADSL申込＆NTT取次キャンペーン**
## 月々わずか780円!!

## ADSL1.5Mも780円で同時キャンペーン中！

フレッツ・ADSL申込&NTT取次キャンペーンについて ◎キャンペーン対象者：2002年4月30日までにフレッツ・ADSL8Mコース、または同1.5Mコースお申込と同時にフレッツ各サービスのNTT取次をアルファインターネットにお申込の方が対象となります。

◎キャンペーン価格有効期間：ご入会月から1年間有効となります。◎キャンペーン価格有効期間終了後は正規料金となります。

### 高速・常時接続サービス（東京エリアから） 入会金無料！
**ブロードバンド フレッツ・ADSL 8Mコース**

月額1,080円（年額11,880円）

1年間キャンペーン価格!!（4月30日までにお申込の方）
月額780円（年額9,360円）

●完全定額制（NTT料金別）●HP容量15MB ●メールアドレス2つ無料(1つは受信専用) ※工事費1000円（初回のみ）

### 高速・常時接続サービス（全国エリア対応） 入会金無料！
**ブロードバンド フレッツ・ADSL1.5Mコース**

月額980円（年額10,780円）

1年間キャンペーン価格!!（4月30日までにお申込の方）
月額780円（年額9,360円）

●完全定額制（NTT料金別）●HP容量15MB ●メールアドレス2つ無料(1つは受信専用) ※工事費1000円（初回のみ）

## お申込はオンラインが簡単！

24時間 オンライン申込受付中
**www.alpha-net.ne.jp**

24時間 資料請求受付電話（音声応答システム） 03-4306-0789
24時間 e-mail資料請求 info@alpha-net.ne.jp
24時間 FAX情報サービス（お申込書の入手に） 03-3376-9898

## テレポケット NTT各種サービス申込も同時受付中！

### ダイヤルアップ接続サービス
**アルファ基本コース** 入会金無料！

月額980円（年額9,800円）

●無制限・使い放題（電話代別）●HP容量15MB ●メールアドレス2つ無料（1つは受信専用）●NTT、KDDI、各種携帯・PHS、モバイル使用可能

《オプション／月額料金》

| | | |
|---|---|---|
| 常時接続 | フレッツ・ISDN | 0円 |
| ブロードバンド | フレッツ・ADSL1.5M | 600円 |
| ブロードバンド | フレッツ・ADSL8M | 700円 |
| モバイル常時接続 | エアーエッジ | 200円 |
| モバイル常時接続 | エアーエッジ128 | 200円 |
| 光ファイバー | Bフレッツマンション | 1,000円 |
| 光ファイバー | Bフレッツファミリー | 2,000円 |

### ドメイン取得ダイヤルアップ接続サービス
**アルファ・ドメインコース**

月額3,580円（年額39,380円）

●無制限・使い放題（電話代別）●独自ドメイン取得 ●HP容量100MB ●電子メールアドレス1つ無料 ●NTT、KDDI、各種携帯・PHS、モバイル使用可能

《オプション／月額料金》

| | | |
|---|---|---|
| 常時接続 | フレッツ・ISDN | 100円 |
| ブロードバンド | フレッツ・ADSL1.5M | 200円 |
| ブロードバンド | フレッツ・ADSL8M | 400円 |
| モバイル常時接続 | エアーエッジ | 100円 |
| モバイル常時接続 | エアーエッジ128 | 100円 |
| 光ファイバー | Bフレッツマンション | 500円 |
| 光ファイバー | Bフレッツファミリー | 1,000円 |

「フレッツ・ISDN」「フレッツ・ADSL」のアクセスポイント！ ただ今、
**全国47都道府県にサービス提供中！**

※「フレッツ・ADSL8M」「Bフレッツ」のサービスエリアは東京のみですが、順次拡大予定。
※一部市区町村にご利用いただけない場合があります。

ダイヤルアップ接続サービスもさらに充実！ ただ今、
**アクセスポイント全国350ヵ所以上！**

詳しくはFAX情報サービス、Webサイトをご利用ください。

［好評につきストラッププレゼント期間延長！］ 2002年5月31日まで
●期間中、アルファインターネットに入会された方にもれなくオリジナル・ロングストラップをプレゼント。

### フレッツISDN常時接続サービス
**常時接続 フレッツ・ISDNコース** 入会金無料！

月額300円（年額3,300円）

●完全定額制（NTT料金別）●HP容量15MB ●メールアドレス2つ無料（1つは受信専用） ※工事費1000円（初回の

### モバイル常時接続サービス（DDIポケットのAirH"専用）
**モバイル常時接続 エアーエッジコース** 入会金無料！

月額980円（年額10,780円）

●モバイル常時接続（DDIポケット「使い放題コース」の場合）●HP容量15MB●メールアドレス2つ無料（1つは受信専用）※工事費1000円（初回のみ）

《オプション／月額料金》
128オプション ……0円

**全国出張サポート受付中**（有料） アルファインターネットの接続から基本講習まで、全国に出張サポートいたします。詳しくはTEL.03-3377-5522までお電話ください。

※すべてのコース・オプションとも入会金無料です。 ※金額は全て税別です。
※各種オプションサービスには初回に工事費1000円が必要です。
●アルファ・ドメインコース以外のコースではホームページの商用利用は出来ません。


## アルファインターネット
インターネット事業部

〒151-0071 東京都渋谷区本町1-28-15
TEL:03-3377-5522 FAX:03-3376-7632
■経済産業省・電子商取引(EC)実証事業契約企業 ■電子商取引推進協議会会員

お問い合わせ 月曜～金曜 11:30～21:00 土曜11:30～17:00 日曜・祝日休み


**TBSとアルファインターネットとの提携によるインターネットショッピング『ishop』展開中！**

### アルファのコンテンツサービス

**アルファの株とマネー** 月額1,500円
プロも使える株コンテンツ！TV東京の信頼できるニュースソースであなただけのポートフォリオが作成できます ●今日の株価材料 ●株の動き ●NYマーケット情報 ●日経平均予想 ●東京マーケット分析 ●東京インタビュー ●オリジナルインタビュー ●TX-クオートサイト

**Art PoPoRo** 無料
本格的バーチャルアート美術館 Art PoPoRo館長・若松裕之、プロデューサー・有森正がニューヨーク、パリ、東京などで活躍中の一流アーティスト20数名に呼びかけ開設したミュージアム

**アルファPAWコース** 月額1,680円
犬専用のコミュニティー「パウ」コンテンツ&ダイヤルアップ接続サービス ●犬の健康、しつけ、映画、エッセイ、ドッグラン、一緒に泊まれるホテル、犬用語解説など豊富なコンテンツ ●無制限・使い放題（電話代別）●HP容量50MB ●メールアドレス3つ無料（1つは受信専用 ●NTT、KDDI、各種携帯・PHS、モバイル使用可能 ●フレッツ・ADSLなどのオプション追加も可能

あなたに、プラス
**αinternet**®
プロバイダ7年の実績が支える信頼のインターネットサービスを提供します。

# 接続0円

## プロバイダへの入会不要

## 電話会社への登録も不要

## 電話料金※3分8.5円だけ

## 新しい「α無料接続サービス」は、入会のプロセスを必要としない、登録や手続きが一切不要なインターネット接続サービスです。

下記電話会社がご利用できます。
NTT・東京電話・KDDI・日本テレコム・九州電話、携帯：NTTドコモ、PHS：DDIポケット・NTTドコモ・アステル（J-Phone、au、Tukaは近日中に対応予定です。）
※3分8.5円はNTT加入電話から同一地区内で接続した場合の料金です。　電話料金は各電話会社の料金に準拠します。
電話料金は契約されている電話会社から請求されます。

（こんな方にはピッタリ）

- 今すぐインターネットを始めたい方
- 登録されるのを好まない方
- カード支払を好まない方
- 気楽に自由に使いたい方

＊このサービスにはe-mail、ホームページのサービスはありません。インターネット接続だけのサービスです。

e-mailを利用されたい方は、
キュリオフリーメール（無料）をお奨めします。
http://www.curio-city.com/

Enjoy The Shopping World! CURiO CiTY http://www.curio-city.com

キュリオフリーメールの「キュリオ占い」は3兆通りの組み合わせから占ったあなたの運勢を毎日無料で配信されます。キュリオシティではショッピング、プレゼントをはじめ、オークション、カレンダー、グリカ、ホットラインなど多彩なメニューがご利用できます。

＊NTTのテレホーダイ、i・アイプラン、エリアプラス、タイムプラスなどのサービスはご利用できません。

### ●下記の電話番号地域でご利用いただけます。（ただいま、日本全国に拡大中!）

下記の電話番号地域以外から接続した場合、市外通話料金が適用されますのでご注意ください。

札幌 011-500-0011　千歳 0123-43-0011　芦別 01242-6-0011　滝川 0125-25-0011　岩見沢 0126-27-0011　栗山 01237-4-0011　石狩 0133-63-0011　小樽 0134-35-0011　余市 0135-30-0011　岩内 0135-60-0011　倶知安 0136-36-0011　伊達 0142-27-0011　室蘭 0143-33-0011　苫小牧 0144-22-0011　鵡川 01454-8-0011　門別富川 01456-6-0011　えりも 01466-7-0011　仙台 022-775-0011　迫 0220-20-0011　岩沼 0223-21-0011　大河原 0224-64-0011　石巻 0225-27-0011　気仙沼 0226-56-0011　築館 0228-28-0011　古川 0229-90-0011　さいたま 048-707-0011　川口 048-277-0011　熊谷 048-510-0011　久喜 0480-30-0011　草加 048-907-0011　川越 049-207-0011　東松山 0493-33-0011　秩父 0494-34-0011　本庄 0495-55-0011　千葉 043-355-0011　市原 0436-33-0011　木更津 0438-88-0011　市川 047-722-0011　館山 0470-37-0011　大原 0470-75-0011　鴨川 0470-91-0011　柏 04-7161-0011　茂原 0475-37-0011　東金 0475-57-0011　成田 0476-76-0011　佐原 0478-77-0011　銚子 0479-27-0011　八日市場 0479-88-0011　東京（23区）03-6777-0011　武蔵野三鷹 0422-75-0011　稲城・国分寺 042-407-0011　立川 042-840-0011　八王子 0426-58-0011　青梅 0428-36-0011　相模原・町田 042-858-0011　横浜 045-555-0011　川崎 044-337-0011　厚木 046-216-0011　小田原 0465-55-0011　平塚 0463-47-0011　藤沢 0466-66-0011　長野 026-466-0011　阿南町 0260-30-0011　大町 0261-66-0011　松本 0263-30-0011　木曽福島 0264-40-0011　伊那 0265-99-0011　飯田 0265-50-0011　諏訪 0266-63-0011　小諸 0267-33-0011　佐久 0267-70-0011　上田 0268-33-0011　中野 0269-55-0011　飯山 0269-66-0011　静岡 054-907-0011　浜松 053-550-0011　掛川 0537-70-0011　磐田 0538-22-0011　天竜 0539-27-0011　清水 0543-30-0011　富士宮 0544-57-0011　富士 0545-50-0011　島田 0547-70-0011　榛原 0548-80-0011　御殿場 0550-71-0011　伊東 0557-66-0011　下田 0558-66-0011　修善寺 0558-80-0011　名古屋 052-225-0011　田原 0531-39-0011　豊橋 0532-22-0011　設楽 05366-6-0011　瀬戸 0561-66-0011　尾張横須賀 0562-22-0011　西尾 0563-33-0011　岡崎 0564-20-0011　豊田 0565-60-0011　刈谷 0566-66-0011　津島 0567-77-0011　春日井 0568-80-0011　半田 0569-30-0011　一宮 0586-66-0011　大阪 06-7222-0011　富田林 0721-27-0011　堺 072-373-0011　岸和田貝塚 0724-42-0011　和泉 0725-25-0011　茨木 0726-56-0011　池田 0727-42-0011　八尾 0729-89-0011　大津 077-570-0011　今津 0740-30-0011　奈良 0742-42-0011　大和高田 0745-50-0011　榛原 0745-90-0011　吉野 07463-7-0011　十津川 07466-5-0011　上北山 07468-8-0011　五条 07472-8-0011　下市 0747-66-0011　水口 0748-80-0011　八日市 0748-50-0011　彦根 0749-44-0011　長浜 0749-77-0011　京都 075-550-0011　亀岡 0771-45-0011　園部 0771-71-0011　峰山 0772-77-0011　宮津 0772-35-0011　舞鶴 0773-73-0011　福知山 0773-30-0011　宇治 0774-84-0011　神戸 078-555-0011　山崎 0790-80-0011　福崎 0790-30-0011　龍野 0791-73-0011　相生 0791-40-0011　姫路 0792-70-0011　加古川 0794-94-0011　三木 0794-71-0011　柏原 0795-83-0011　西脇 0795-33-0011　浜坂 0796-86-0011　豊岡 0796-33-0011　西宮 0798-88-0011　津名 0799-66-0011　洲本 0799-21-0011　広島 082-552-0011　呉 0823-37-0011　三次 0824-42-0011　東広島 0824-96-0011　庄原 08247-6-0011　千代田 0826-62-0011　安芸吉田 0826-48-0011　加計 08263-4-0011　廿日市 0829-52-0011　竹原 0846-47-0011　府中 0847-58-0011　甲山 0847-28-0011　東城 08477-8-0011　尾道 0848-33-0011　福山 084-998-0011　福岡 092-233-0011　前原 092-333-0011　行橋 0930-35-0011　北九州 093-888-0011　宗像 0940-22-0011　久留米 0942-25-0011　八女 0943-55-0011　田主丸 09437-8-0011　大牟田 0944-70-0011　甘木 0946-33-0011　田川 0947-33-0011　飯塚 0948-33-0011　直方 0949-30-0011

### ●下記設定で今すぐにでもインターネットに接続OK!

接続・設定情報

プライマリーDNSアドレス
**61.195.80.9**

セカンダリーDNSアドレス
**61.195.80.10**

| ユーザーID | パスワード |
|---|---|
| a@a | aa |

### ●アルファインターネット「α無料接続サービス」についてのお問い合わせは

www.alpha-net.ne.jp
e-mail:0011@alpha-net.ne.jp
tel:03-3377-5522

＊本サービスの技術サポートは提供しておりません。


アルファインターネット 通信事業部
〒151-0071 東京都渋谷区本町1-28-15
TEL:03-3377-5522 FAX:03-3376-7632


■経済産業省・電子商取引（EC）実証事業契約企業 ■電子商取引推進協議会会員

お問い合わせ 月曜〜金曜 11：30〜16：30 土曜・日曜・祝日休み

あなたに、プラス
αinternet®
プロバイダ7年の実績が支える信頼のインターネットサービスを提供します。

# マイコン BASIC Magazine

http://www.basicmagazine.net

# 5 2002 MAY

巻頭カラー企画

編集部の学校訪問記

進路や適正に合わせた時間割で，自覚を持った学習を可能にする
**県立神奈川総合高等学校** 6

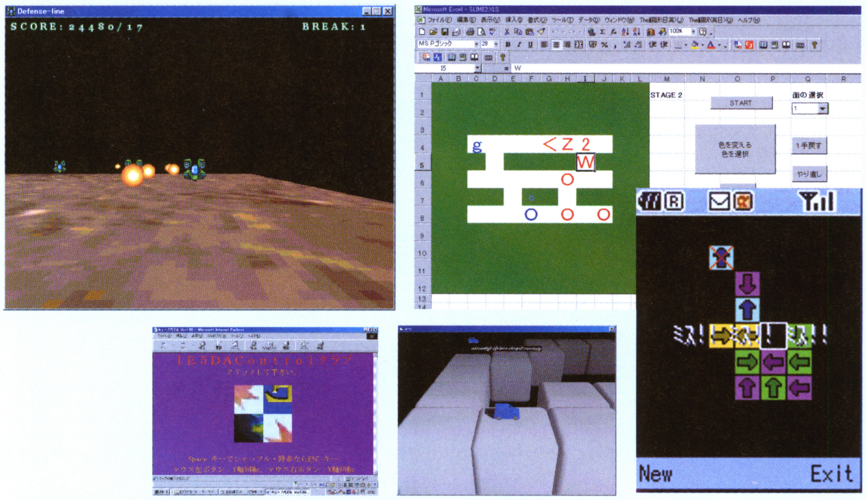
独自のテキストを駆使しながら情報教育にがんばる先生，それに応える生徒たち
**横浜市立本郷中学校** 8

好評連載

見やすい，楽しい，誰でも使えるサイトを実現！
**一歩先ゆくホームページ作り**
**文字・文章——いろいろな書き方・見せ方** 10

特選投稿プログラム・コーナー 45

| | | | |
|---|---|---|---|
| Visual Basic V6.0 | 攻めてくる数字軍団 | 有田安彦 | 51 |
| Visual Basic V6.0 | ナンバープレイっす | 米久保直樹 | 56 |
| Visual Basic V6.0+DirectX | LaserShootⅣ | 伊藤隆朗 | 58 |
| Visual Basic V6.0+DirectX | a-z | 橘一馬 | 60 |
| VBScript + DirectX | キューブパズル | 金井義男 | 62 |
| Excel(VBA) | パニックスライム移植版 | 内田寛 | 67 |
| F-BASIC V6.0 | 感撃！イロイロ棒 | MONOMON | 71 |
| HSP V2.55 | Defense-line | くろすけ | 78 |
| Javaアプレット | vegetable3 | 長谷川誠 | 82 |
| iアプリ | Pair Arrow | 峠恒司 | 88 |
| PIECE | Nazo(ナゾ) | 谷澤友哉 | 93 |


**掲載「投稿プログラム言語」入手方法と入力手順** 46

今月のベスト・プログラマーは誰の手に！
**明日のスタープログラマー** 97

今月のベスト・プログラムを解説！
**Dr.Dのベスト・プログラム研究室** 98

© マイコンBASICマガジン5月号
第21巻 第5号 2002年4月25日印刷
2002年5月1日発行
定価 550円（本体524円）
電波新聞社 2002 Printed in Japan
編集発行人：平山哲雄
東京都品川区東五反田1-11-15
郵便番号 141-8715
電話 東京(03)3445-6111(大代表)
振替（東京）00150-3-51961
印刷：奥村印刷(株)　製本：(株)堅省堂


<本誌掲載の記事，プログラムの無断使用を禁止します>

CPUの歴史と未来をさぐる！
**モバイルプロセッサがユビキタスな世界を作る！** 21

BM基礎講座
ベーシックプログラミング教室
**第2回 分岐処理と繰り返し** 29

はじめようMicrosoft OfficeでBASICプログラミング
**第1回 WordやExcelでできる** 35

PCハードウェアテクニック
**旧型AT機を完全復活**
**マザーボード換装テクニック** …… 39

マンガで学ぶ！コンピュータの世界
**パソコンレクチャー　ザ・ビギニング**
**バックアップも忘れないで！** …… 100

これから始めるブロードバンド活用術
**HowTo!! ADSL体験記**
**「eAccess」接続サービスでの設定方法** …… 104

もっと身近にプログラミング
**チャレンジ！オリジナル・プログラム言語の制作！**
**その4 ジャンプには，ついてまわるよコンバータ** …… 109

フリーソフトHSP完全マスター　楽々3Dプログラミング！
**第4回 MXモデルでゲーム制作（1）** …… 114

教科「情報」など授業にも使える！ Visual Basicアルゴリズム講座
**第2回 速度の計算をしてみよう** …… 117

21世紀の遊びはコレ！　ロボ道場
**AIBOを無線LANでコントロール！** …… 122

時代を先取り！目指せ最先端プログラマー！ .NETプログラミング入門
**第2回 .C#言語（前編）** …… 126

電子工作にチャレンジ！　自作一代記
**AKI-PIC877モジュール用ベースボード・キット（その2）** …… 131

読者のコーナー
**あくせすROOM／プレゼント** …… 134

ソフト・メーカーのPRのページ
**SOFTWARE MAKER HOT INFORMATION** …… 135

***・6月号は5月8日発売です！・***

広告索引

**ハード**
ソニー　表4
**プロバイダ**
アルファインターネット　表2〜前付カラー3
**ソフト**
日本ファルコム　表3

**学校**
東京ゲームデザイナー学院　後付137
**パソコンショップ**
若松通商　記事中44
秋月電子通商　記事中133
パシフィックネット　後付138

**好評発売中!!**

楽しく学習できる
**新まんが電気学入門**

監修：江田　稔
（文部科学省初等中等教育局
漫画：原子　力
●A5判／247頁　送料 380円
定価1,500円（税込）

電気のはたらきから
パソコンのしくみまで
わかりやすくまんがで解説！

主な内容

●第１章／静電気
静電気（まさつ電気）／正の電気と負の電気／はく（箔）検電器／静電気と電流／雷とは？ ……他

●第2章／乾電池と豆電球
電池／電池の原理／乾電池のしくみ／豆電球のしくみ／電気を通すもの、通さないもの／導線のつなぎ方／配線図と回路記号／豆電球の直列つなぎ ……他

●第3章／電流と電圧
閉回路と開回路／電流／電荷／電圧／抵抗（電気抵抗）／抵抗のいろいろ／抵抗値の見分け方／オームの法則／電流計（直流用）電圧計（直流用）のしくみ ……他

●第4章／回路計算
電圧の総和／電流の総和／電圧のかかり方／抵抗の直列接続と合成抵抗値／合成抵抗の中の未知抵抗／内部抵抗の求め方／キルヒホッフの（第１）法則 ……他

●第5章／電気エネルギー
電流による発熱作用／電流による発熱原理／発熱量の測定／ジュールの法則／抵抗の接続方法と発熱量の大きさ・電力／電力量／積算電力計（交流用） ……他

●第6章／磁石と電磁石
磁石と磁極／Ｎ極とＳ極／磁力の大きさ／磁気誘導／Ｎ極とＳ極の分離／磁力線／地磁気／電流の大きさと方位磁針の振れ／右ねじの法則 ……他

●第7章／電磁誘導
電磁誘導／レンツの法則（誘導電流の向き）／ファラデーの電磁誘導の法則／コイルの巻き数と誘導電圧／電気ブランコによる電磁誘導 ……他

●第8章／電流と電子
火花放電／真空放電とグロー放電／蛍光灯と電球／クルックス管と陰極線／原子と電子／電子と火花放電／トランジスタの構造 ……他

●第9章／パソコン
パソコンって何？／パソコンでどんなことができるの？／パソコンはどう動くの？／パソコンのいろいろな形態／パソコンの簡単な歴史／パソコンの仕組み—CPU・メモリ・補助記憶装置・フロッピーディスク・ハードディスク・CD-ROM／周辺機器—キーボード・マウス・ディスプレイ・プリンタ・スキャナ・デジタルカメラ（デジカメ）／OSの働き—OSって何？／OSの働き—OSの種類は？／日本語入力ソフト／アプリケーションソフト／LAN（ラン）て何？／USBとは？／パソコン関連の単位／インターネットとは？／インターネットはどうしてできたか？／インターネットサービスプロバイダ／インターネットで何ができるか？／WWWとは？／電子メール

電波新聞社
東京本社　電話（03）3445-8201（ダイヤルイン）
〒141-8715　東京都品川区東五反田　1-11-15 電波ビル

「情報A」を受講していた生徒のみなさん

## 編集部の学校訪問記

1995年開校の神奈川総合高等学校は，神奈川県で初めて単位制を取り入れた全日制・普通科高校として注目されています。単位制とは，1年，2年，3年といった学年制をとらず，時間割は生徒各自が，興味や関心，進路にあわせて組むことができるというもの。ですから授業も，年齢の異なる生徒が一緒に学習しています（通常の学校でいう1年から3年が入り混じって学習している）。

校舎も近代的で，大学と見まちがうほどです。中央棟と呼ばれる10階建て校舎の6階が，通称「コンピュータと語学の階」。コンピュータ学習室などが設置されているフロアです。今回は，ここでの授業風景を中心に取材してきました。

# 神奈川県立神奈川総合高等学校

http://www.kanagawasohgoh-h.ed.jp/

## 進路や適正に合わせた時間割で，自覚を持った学習を可能にする

### 全国でも例を見ない単位制を取り入れた全日制・普通科の高校

新構想の高校を開設しようということで，神奈川県の教育庁に準備室が設置されたのが1993年。1995年4月に神奈川総合高等学校として開校されました。

特徴は，神奈川県で初めて単位制を基調とした普通科高校ということ。自分で時間割を作成し，クラスや年齢の異なった生徒と一緒に学習，3年間で80単位以上を修得して卒業していきます。

選択できる科目も多様で，高校の普通科に置かれている科目の他，文学系・社会系・環境系・数学系・バイオ系・化学系・スポーツ系・音楽系・美術系・舞台系・福祉系・工学系・情報系の13類型が設けてあり，進路や適正に合わせて自由に組むことができます。

また，工学系などは同じ敷地内に隣接している神奈川工業高校の教室を使用するなど，学校連携も行われています。

募集枠は，海外帰国子女や神奈川県に住む外国人生徒，留学生の受け入れ態勢もあり，国際色も豊かです。海外での教育と同校の教育がスムーズに行えるよう，4月から9月までの前期，10月から3月までの後期という2学期制も特徴の一つで，4月入学のほか，10月入学の制度があります。

### コンピュータ設置フロアには機能別に四つの教室が

10階建て校舎の6階が通称「コンピュータと語学の階」と呼ばれているフロアです。この階にあるコンピュータ教室は，

▲10階建て校舎の中庭に立つオブジェ

①コンピュータ学習室A……COMPAQ Prosignia Desktop（Windows2000）40台

②コンピュータ学習室B……東芝PV3000（Windows95）20台→5月更新予定

③マルチメディア学習室……富士通FMV6868CLT（Windows2000）40台

④LL学習室A……東芝EQUIUM（WindowsNT）30台

という構成です。

▲情報系の吉田健一先生。授業もいろいろ工夫しながら展開してる

▲情報系の吉田史明先生。各教室を案内していただいた

▲コンピュータ学習室A

### 「情報A」の授業

▼「1年間の授業を振り返って」と題したプレゼンテーションをPowerPointで作成中。友達同士で意見交換しあったり，先生に質問を浴びせたり，90分があっという間

▲コンピュータ学習室B。数学用ソフト「Mathmatica」を使用した数学や理科などの授業に使われるほか，放課後などは生徒に開放している

▲マルチメディア学習室。映像と音楽中心の授業が行われる教室。40台すべてにPaint ShopProがインストールされている他，そのうちの20台にはプレミアが，残り20台にはRoland SC8850が！

▲神奈川総合高校ホームページ。生徒が組んだ時間割例なども公開されている

◀LL学習室A。外国語の授業で主に使用する

## PowerPointでプレゼンテーション！「情報A」の授業拝見

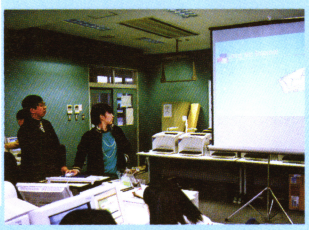
今回は，コンピュータ学習室Aで行われていた必修科目「情報A」の授業を拝見しました。(教科「情報」は2003年4月から全高校で必修科目となる。現在は前倒しで行っている高校も多い)

授業は31人講座で，うち1年生が23人でした。内容は，“1年間の「情報」の授業を振り返って”と題したPower Pointでのプレゼンテーション作成に取り組んでいるところで，計4時間（1時間90分授業）の3時間目，グループ発表の仕上げに取り掛かっているところでした。

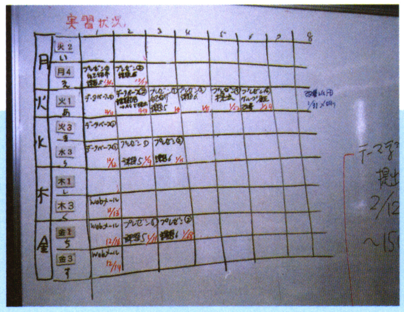
指導している吉田健一先生は「すでに文字入力や基本的なことはやり終えているので，表現をどうするかデザインセンスなども試されます」と話していました。生徒は，各自趣向を凝らし仕上げています。

**「情報A」昨年度１年間の大まかな流れ**

- Wordを使用した文字入力の基本
- Excelを使用したグラフ作成
- 知的所有権／モラルの学習
- Outolook Expressでのメール設定
- Power Pointを使用したプレゼンテーション



◀課題の出来上がった生徒は実際にプロジェクタに映し出していた。「パソコンのモニタとプロジェクタでは色も変わりますね」（吉田健一先生）



▲コンピュータ学習室Aでの実習時間割

## コンテストや検定試験で実力を見せ始めている

昨年の経済産業省主催プログラミング・コンテストでは，同校の水谷正慶くんが優秀賞を受賞しました。個人で応募したということですが，この受賞も，自らの個性にあった授業を選択することで，さらにその個性を伸ばすことができる制度の実践例と言ってもいいでしょう。

他にも，「コンピュータ利用技術検定」「システムアドミニストレータ試験」「基本情報処理技術者試験」「情報処理検定」など各種検定試験に合格者を出しているなど，専門的な学習が入学後早い時期から始められるという点も注目です。

**平成13年度全国高校生・専門学校生プログラミング・コンテスト**

**プログラミング部門（高校生の部）優秀賞**

水谷正慶くん
（当時3年生）
予約管理システム
「かもめ」

図書館などの施設を利用する際の予約管理ソフト。オンライン化によって，どこからでも予約登録・閲覧が可能なシステム。

神奈川県立神奈川総合高等学校　〒221-0812 横浜市神奈川区平川町19-2　☎045-491-2000

コンピュータコース選択の（取材当時2年生の）みなさん

編集部の学校訪問記

# 横浜市立本郷中学校

http://www.edu.city.yokohama.jp/jhs/hongou-jhs

## 独自のテキストを駆使しながら情報教育にがんばる先生と，それに応える生徒たち

横浜市の都心部から電車で20～30分，市の南側に位置し，鎌倉市と隣接している栄区，区役所や消防署など公共施設が隣接した区の中心に本郷中学校（横山勝洋校長）は位置しています。

創立は昭和22年（1947年）と古いながら，同校があるJR本郷台駅は，公団住宅と市営住宅が建ち並び，住民も新旧さまざまで，生徒たちも新旧のバランスがうまくとれているようです。

今回は，2年生のインターネットやホームページ作りの選択授業を拝見してきました。

### パソコン室の環境は？ －ノートPCと光回線が！－

本郷中学校は，情報教育に熱心な学校です。「入学してくる生徒のパソコン所有率は，1999年度35％，2000年度49％，2001年度75％と急激に伸びています。一般家庭へのコンピュータの普及に伴って，情報社会に対応できる生徒の育成は，私どもに課せられた使命だと思います」と語るのは，今回案内をしていただいた白川仁美先生。白川先生のがんばりが，同校の情報教育推進の原動力でもあります。

同校のパソコン室には，IBMのノートPC「ThinkPad」（CPU：Intel Celeron）が40台（生徒用），OSはWindows98SEです。インターネット環境は，これまでのISDNに加え，文部科学省・総務省連携の「学校インターネット3」事業のモデル校として，2001年12月に光ファイバーケーブル接続も可能になりました。

校内LANの構築は，白川先生を中心にした先生方の手作りです。秋葉原に行って，HUBやLANケーブルなど4万7千円で購入してきたのだそうです。

### 競争率が高い選択授業の「コンピュータコース」

パソコン室は，技術・家庭科の情報基礎領域を中心に，社会科，理科や総合的な学習などの教科では調べ学習として利用されています。選択教科のコンピュータコースの授業では，2・3年生がホームページ制作に取り組んでいます。今回，拝見したのは2年生の授業でした。友達同士教えあったり，相談しあったり，または先生を呼んで直接指導してもらったり，さまざまです。

選択教科は1コース定員20名のとこ

▲パソコン室のようす。パソコンは「ThinkPad」が40台，プリンタは「Canon　BJ F300」が20台，整然と並べられている。先生がつくことを条件に放課後も開放している

◀選択授業コンピュータコースでホームページ制作に取り組む生徒たち

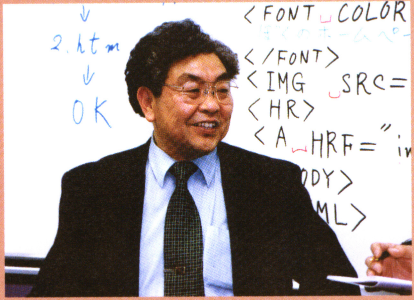
ろ，コンピュータコースだけは，毎年5倍の倍率のなか抽選で選ばれるそうです。それでも定員オーバーの24名で構成されていました。

教材は，白川先生のノウハウが小冊子としてまとめられている「INTERNET 超初心者のためのホームページ作り」（総95ページ）。「6年前，自分でホームページを作成しようとしたのですが，当時テキストもなく，独学でHTMLを学びました。2か月で完成したのですが，本当に苦労しました」という白川先生力作のテキストです。

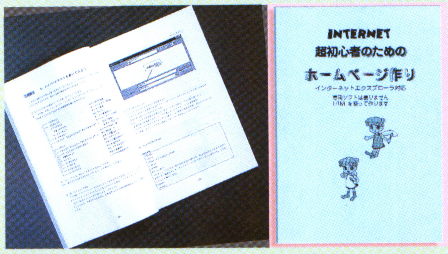
Windowsの“メモ帳”を使ってHTML言語で作成していく手順が親切ていねいに記述されています。

「生徒の表現力を限られた時間内で引き出すには，ホームページ制作は効果的です。生徒の達成感も感じ取れます」（白川先生）。

## 情報教育担当・白川仁美先生




▲同校の情報教育にはかかせない白川先生。「以前はキーボードからの入力などの時間をさいていたが，今の生徒はほぼ入力はできる。即実習に入れる」と語る




▲白川先生手作りのテキスト「INTERNET 超初心者のためのホームページ作り」。他校の授業や市などの研修会でも活用されている

コンピュータに関しては10年以上も前から中学生に指導してきました。基本はプログラム学習です。以前はBASICを通して行っていたものが，今はHTML言語になっています。HTMLも書いた順番に表記されていくという意味では，立派なプログラムです。扱いも簡単で生徒も表現力が増してきたように思います。

こうした生徒の活動のようすや作品などもホームページ上で公開できるようにしたいと考えています。

### 選択授業「コンピュータコース」の大まかな流れ

**2年** 1学期…HTMLの基礎を学ぶ
2学期…自分でホームページを探す。真似をする

**3年** テーマに沿って自分のホームページを作り上げる

※生徒の自主性にまかせた進めかた

## コンテスト出展も目標に！

横浜市や神奈川県など主催のさまざまなコンテストへの参加も大いに刺激になっているようです。HTMLを勉強して約6か月の作品が，横浜市のコンテストで入選も果たしています。

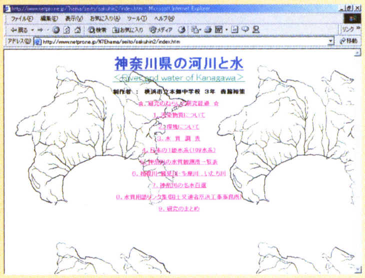
### 国際コミュニケーション基金主催「第4回ICFホームページコンテスト」優秀賞

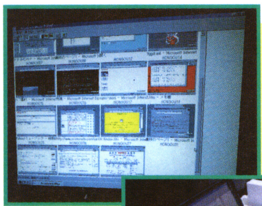
**【神奈川県の河川と水】**

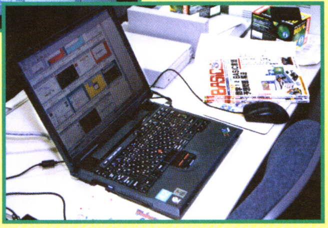
**森脇裕策くん（当時3年生）**

http://www.netpro.ne.jp/%7Ehama/seito/sakuhin2/index.htm

理科の選択授業で調べたテーマをホームページにまとめたもの。白川先生の教えもあり，はじめてのHTMLでも完成度の高いものになっている。







◀先生用パソコンではAcademicWareで生徒の画面が一度にモニタできる

## 本郷中学校パソコン室で使用されているソフト

ハードディスク・キーパー
**HDD KEEPER**

ハードディスク・データの誤った追加や削除，設定変更，フォーマットなど，あらゆるデータが変更されても，再起動するだけで完全に元の状態に戻すことができるソフト。
トーエイ工業
http://www.to-ei.co.jp/top.html

サイバー・シッター
**CYBERsitter**

スタンドアロン型のインターネット・フィルタリング・ソフト。有害情報の中に含まれる「語句」などを高速で検索し，情報全体の有害性を判断する。
アイキュエス
http://www.iqs-j.com/

アカデミック・ウエア
**AcademicWare**

ネットワーク機能を使って教師と学生の画面を双方向で通信。教師用パソコンから直接生徒の指導が行える。複数の生徒の画面を同時にモニタでき，教師用端末から遠隔操作も可能。
コンパック
http://www.compac.co.jp/

**横浜市立本郷中学校** 〒247-0005 横浜市栄区桂町84-14 ☎045-892-2155/2156

**見やすい，楽しい，誰でも使えるサイトを実現！**

# 一歩先ゆくホームページ作り

## 第2回　文字・文章――いろいろな書き方・見せ方

白木ミホ

**マシンやブラウザ，回線の種類や新旧を問わず，誰もが快適に見られる。そんなユニバーサルデザインのホームページを作ろう！ということでスタートした本講座。今回は基本にして極意ともいえるテキストの世界を極めてみましょう。本誌読者のキミならば，作成支援ソフトで「なんとなくできちゃった」なんてホームページとはオサラバしなくちゃね！**

### ホームページの基本はテキスト

ひと昔，といってもほんの数年前まで，ホームページといえば，タグと呼ばれるhtml言語を地道に打ちこんで作る，非常にマニアックなものでした。それが，一部マニアの手を離れて一般の人，それも小学生からお年寄りまで，さまざまな人たちに知られ，実際に制作・発表されるようになったのは，ひとえに「ホームページの作成支援ソフト」が登場したおかげです。これによって専門的な知識のない人にもホームページは簡単に作れるようになりました。

さらに，これまたひと昔前まで，ホームページといえば，単色の背景に文字がズラズラ，たまーに小さな画像がポチッとついているというのがほとんどでしたが，最近では大きなサイズの凝ったＣＧ画像や，デジカメで撮影した写真，はたまたアニメーションから音楽まで，その内容もずいぶん派手に，多様になったものです。こちらももちろん作成支援ソフトの影響大ですが，加えて，それだけの情報を高速で処理できる通信回線と，パソコン本体の性能アップが大きく貢献しているわけです。

ところが，「いろいろできる」ということは，どうも「いろいろやらないと損かも」という妙な理屈を生みがちです。したがって新規にホームページを作ろうと勢いこんでいる人は，どうも「文字だけのサイトは地味」という妙な思いこみを抱きがちです。

でもそんなことはないのです。むしろホームページの王道は「テキストサイト」，つまり文章を主体とした内容にこそあるのだと，先月号の本講座でも紹介しました。現にアクセスランキングなどで上位に食いこむものはテキストサイトが多いですし，誰にでも書ける自己紹介と日記は，ホームページの入口でありながら，それだけに見せ方を要求され，上級者たちがしのぎを削っている場所でもあるのです。さらにテキストには，ただつらつらと表示するだけでない，多彩な見せ方が意外とあります。たかが文字列，されど文字列なのです。

さらに，その作り方に「ポリシー」をもったホームページ，すなわち誰にでも見やすい・使いやすいユニバーサルデザインのホームページを作る。そのために，ぜひ覚えておいてほしいこともちろんあります。

### 文字列・文章の表示～基本

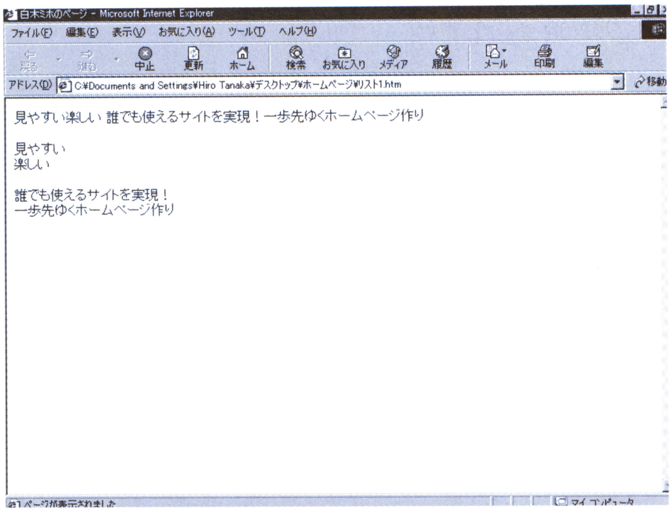
まずはホームページにおける文字列表示のいちばんの基本をタグから理解しておきましょう。

タグは，一部の例外を除いて「ここからここまで」という範囲指定をする言語です。ですから，<xxx>と書いたら必ず</xxx>と，終わりを意味する「/」（スラッシュ）のついたタグを忘れないで下さい。

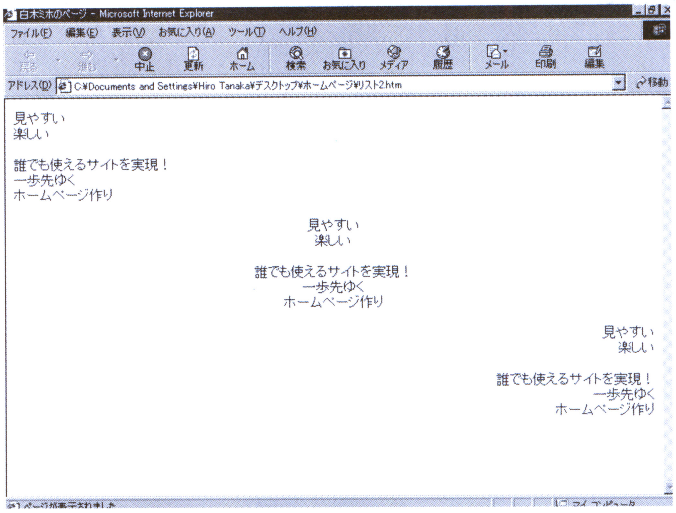
#### 改行，改段落
<br>，<p>

htmlテキストファイルに文章を書くとき，ファイル上の見た目では改行されていても，ブラウザには認識されません（**第1図・リスト1**）。

改行には改行のためのタグ<br>を書きこみましょう。これは「/」を伴わずに単独で使用できるタグです。また，段落と段落の間に１行分の空白を入れるタグ<p>は，本来は「ここからここまでが第１段落ですよ」とくくって，次のくくりとの間に１行分の空白を入れる，というものでしたが，今やただの？「１行あける」記号と化し，</p>がなくても動作に




《第１図》改行・改段落はタグで！テキストファイル上で改行してもブラウザには反映されないのです。注意！




《第２図》左寄せ・中央寄せ・右寄せ
ホームページの雰囲気を決める最重要タグかも

《リスト1》第1図のソースリスト

```
<HTML>
<HEAD>
<TITLE>白木ミホのページ</TITLE>
</HEAD>
<BODY>

見やすい
楽しい

誰でも使えるサイトを実現！
一歩先ゆくホームページ作り

<p>
見やすい<br>
楽しい<br>
</p>
<p>
誰でも使えるサイトを実現！<br>
一歩先ゆくホームページ作り<br>
</p>

</BODY>
</HTML>
```

《リスト2》第2図のソースリスト

```
<HTML>
<HEAD>
<TITLE>白木ミホのページ</TITLE>
</HEAD>
<BODY>
<p>
見やすい<br>
楽しい<br>
</p>
<p>
誰でも使えるサイトを実現！<br>
一歩先ゆく<br>
ホームページ作り<br>
</p>
<center>
<p>
見やすい<br>
楽しい<br>
</p>
<p>
誰でも使えるサイトを実現！<br>
一歩先ゆく<br>
ホームページ作り<br>
</p>
</center>
<p align="right">
見やすい<br>
楽しい<br>
</p>
<p align="right">
誰でも使えるサイトを実現！<br>
一歩先ゆく<br>
ホームページ作り<br>
</p>
</BODY>
</HTML>
```

影響しなくなっているという，ちょっと不思議な指定のタグです。

## 左揃え，右揃え，中央揃え
<p align="　">，<center>

画面全体のどこに文字列を揃えるか，つまり**第2図**のようなレイアウトを指示するのが「左揃え，右揃え，中央揃え」です。何も指定がなければブラウザの表示は左揃えになりますが，これを右に指定する場合は<p align="right">とタグを書きます。中央に揃える場合は<p align="center">と書くのがよりよいのですが，今は独自のタグの<center></center>を使うことの方が多いかもしれません（**リスト2**）。**第1図**と**第2図**では同じ文字列を使っていますが，雰囲気はだいぶ変わりますね。

あえて書くなら「段落の左揃え」を表す<p align="left">という指示も可能です。

## htmlテキストの禁じ手3種

キミのホームページがヒンシュクを買わないために，htmlテキストで文字を表示する場合の「禁じ手３種」も押さえておきましょう。

一つ目は，①②などの丸付き数字や，ⅠⅡのようなローマ数字，ハートマークなどの特殊記号といった，いわゆる「機種依存文字」です。使う機種によって変換時のコード（文字に与えられた番号）が違っているため，まったく違う文字が出てきてしまったり，あるいは該当文字がなかったりする，やっかい者です。意味不明な記号を出現させないよう，現状ではホームページ上では使わないというのがルールになっています。といっても記号が全部ダメというのでなく，○●や

□■，◆★☆♪※などは安全に使えます。ナゾといえばナゾですが，現状ではそうなんだと覚えておいてください。

二つ目は，文字化けの原因となる「半角カナ」です。最近はｉモードユーザーがいつものクセでつい半角カナを使ってしまい，盛大に文字化けを起こしていることがありますので注意しましょう。

そして三つ目は，前回も触れましたが「空白」の使い方です。たとえば「自␣己␣紹␣介」のように，デザインとして一つの単語の中に空白を入れているのをよく見かけますが，目の不自由な人たちが使用している「音声リーダ」は，それらを1単語として認識できないために誤読をしてしまうのです。これはユニバーサルデザインのホームページを目指すキミなら当然自粛してほしい点です。

## 色の話

実際にホームページを作ろうと思うと，意外と悩んでしまうのが「色」ではないでしょうか。ホームページ全体の印象を決めてしまう色使いは，作り手のセンスがいちばん表れるところです。カッコいいページもダサいページも，この色使いでかなり決まってしまいますし，格好以外にも見やすさという実利面で，好評あるいは不評を買う分かれ目となる，それが色使いです。

### 大原則は「明暗」

色使いにおいていちばん大事な原則は，文字と背景に明暗の差をはっきりとつけることです。

そんなの当たり前だろうと思うかもしれませんが，実際に公開されているホームページ上では，これがかなり忘れられているのが実態です。凝り過ぎたためにかえって見にくくなってしまっている場合が多いようです。

たとえば原色の赤の背景に同じ原色の青の文字。色ははっきり分かれていますが，どちらも同じ原色で「強い明るさ」同士のため明暗の差がない，はっきりしているのに読みにくい例の代表と言えます。同じ理由で黒い背景に赤い文字。これも赤が沈みやすい色なので「暗い色」同士となり，見えづらくなります。

反対にパステルカラーの背景にパステルカラーの文字は「薄い色」同士で紛れやすい例。黄色と黄緑などは「近い色合い」同士でこれも見分けづらくなります。

見て分かりやすい色使いとは，暗い色に対して明るい色，または薄い色に対して濃い色，というように対比がしっかり離れているものなのです。

そして明と暗，濃と淡がいちばん大きく開いている2色とは，白と黒なのです。

### 禁じ手と注意点

ユニバーサルデザインという面で考えた場合，色は大変重要です。私は色覚はオッケーだよ，なんて考えるのは甘いっ，それでなくてもパソコンのモニタは目が疲れるのです。そこにわざわざ見にくい色使いを持ってきて，疲れ目のネットサーファーたちにヒンシュクを買っては，せっかくマイ・ホームページを立ち上げてもアクセス数が伸びません。ましてや自分の管理するホームページで目がチカチカするのはなおいただけません。

人間の目の色覚の特性上，「原色の赤と緑」の組み合わせは特に見えづらい配色の代表です。中にはまったく見えなくなってしまう人もいますので，禁じ手とするべきでしょう。ところが，クリスマスのシーズンに，この最も見にくいはずの原色の赤と緑のコーディネートはかなり多く出現します。ユニバーサルデザインのホームページ作りとしては，同じ赤と緑であっても原色同士は避けて，一方をパステルカラーにするなどの配慮をするとよいでしょう。

さらに，使用マシンによる差として覚えておきたいのは，ウィンドウズよりマッキントッシュの方が色が明るい，という事実です。ときどき妙に暗いホームページを見かけますが，もしかしたらその管理人はマックユーザーかもしれません。逆にマックユーザーにしてみれば，自分のマシン上では問題ないと思っていた配色が，ウィンドウズ機で見たらやたら暗くてガク然，ということもよく起こっています。

ブラウン管か，はたまた液晶かというモニタの違いもあるようです。私自身の目には液晶画面の方が白っぽく見えるのですが，皆さんの目にはどう映っているでしょうか。

《写真1》Mariのいろえんぴつ
http://member.nifty.ne.jp/marimari/index.htm
**ブラウザのレイアウトと色の組み合せをトータルで見られる便利サイト。残念ながら誌面では再現できませんので，ぜひ実画面を見てください。探しやすく見やすい嬉しい構成です**

### セーフカラー

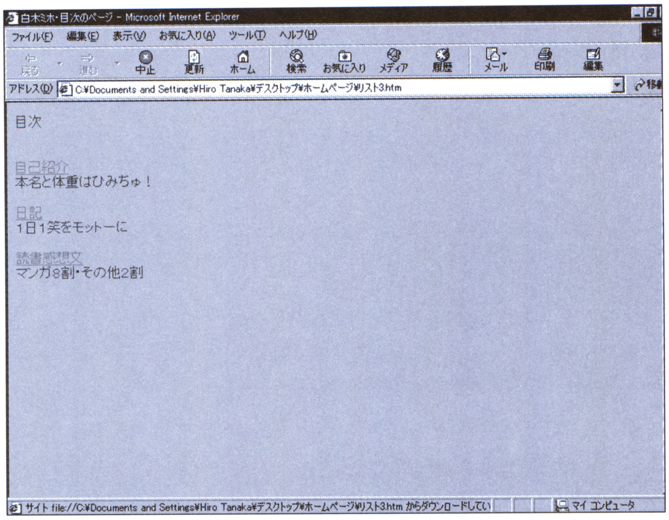
そんな風に，見る側の主観によって大きく変わってしまう色の世界ですが，ホームページ上で色を指定する方法は現在2種類あります。

一つはredやblueといった英語名をそのまま使う色名指定，そしてもう一つは0〜9の数字とa〜fのアルファベットの合計16個を使う16進法のセーフカラー指定です。

色名指定はタグを打ちこむ際に分かりやすくてよいのですが，もともとはインターネット・エクスプローラの機能で，のちにネットスケープにも有効になったもののため，実際はユニバーサルデザインには向きません。

16進法はその点で「セーフ」カラーと呼ばれています。最初の2桁がR＝赤，次の2桁がB＝青，最後の2桁がG＝緑をそれぞれ示し，光の三原色を数値で設定します。#000000が黒，#ffffffが白となります。ホームページ作成支援ソフトでは，色見本表で指定すると自動的にこのセーフカラーの数値がタグに取りこまれるものが多いですが，自分でタグを打つときは画像系ソフトで確かめたり，あるいはカラーコーディネートを紹介しているホームページで色見本の数値をコピーしてくるのも便利でしょう（**写真1**）。

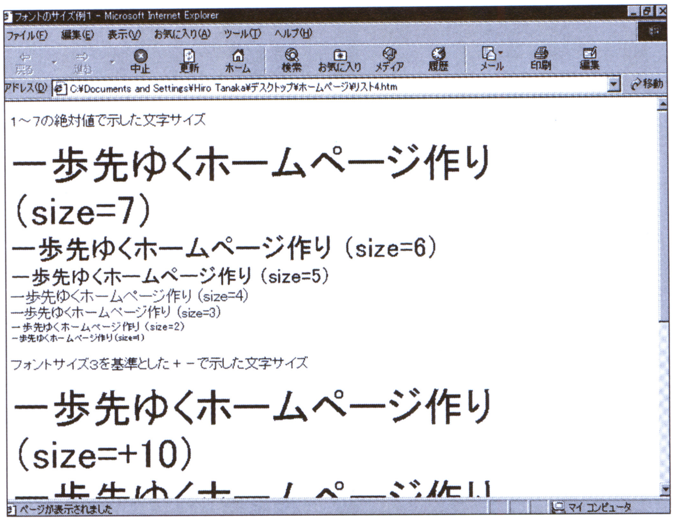
ちなみに色をタグで指定しなかった場




《第３図》背景とページ全体の文字色を指定。他ページとリンクしている文字列の色が変わります…見えるかな？

《リスト3》第3図のソースリスト

```
<HTML>
<HEAD>
<TITLE>白木ミホ・目次のページ</TITLE>
</HEAD>
<BODY bgcolor="#FFCCCC" text="#800040" link="#ff0000"
vlink="#000099" alink="#0000ff">
<P>目次<BR>
<BR>
<BR>
<A href="self.html">自己紹介</A><BR>
本名と体重はひみちゅ！<BR>
<BR>
<A href="diary.html">日記</A><BR>
１日１笑をモットーに<BR>
<BR>
<A href="books.html">読書感想文</A><BR>
マンガ８割・その他２割</P>
</BODY>
</HTML>
```




《第４図》一部の文字色と文字サイズを指定。タグは長いけれどよく見ればカンタン。大きさはブラウザによっても変わりますが全部で7段階というのは共通です

《リスト4》第4図のソースリスト

```
<HTML>
<HEAD>
<TITLE>フォントのサイズ例１</TITLE>
</HEAD>
<BODY bgcolor="#ffcc99" text="#000000" link="#ff0000"
vlink="#000099" alink="#0000ff">
<P>１～７の絶対値で示した文字サイズ<BR>
<BR>
<FONT size="7">一歩先ゆくホームページ作り （size=7）</FONT><BR>
<FONT size="6">一歩先ゆくホームページ作り （size=6）</FONT><BR>
<FONT size="5">一歩先ゆくホームページ作り （size=5）</FONT><BR>
<FONT size="4">一歩先ゆくホームページ作り （size=4）</FONT><BR>
<FONT size="3">一歩先ゆくホームページ作り （size=3）</FONT><BR>
<FONT size="2">一歩先ゆくホームページ作り （size=2）</FONT><BR>
<FONT size="1">一歩先ゆくホームページ作り （size=1）</FONT><BR>
<BR>
フォントサイズ３を基準とした＋－で示した文字サイズ<BR>
<BR>
<FONT size="+4">一歩先ゆくホームページ作り （size=+10）</FONT><BR>
<FONT size="+4">一歩先ゆくホームページ作り （size=+5）</FONT><BR>
<FONT size="+4">一歩先ゆくホームページ作り （size=+4）</FONT><BR>
<FONT size="+3">一歩先ゆくホームページ作り （size=+3）</FONT><BR>
<FONT size="+2">一歩先ゆくホームページ作り （size=+2）</FONT><BR>
<FONT size="+1">一歩先ゆくホームページ作り （size=+1）</FONT><BR>
<FONT size="+0">一歩先ゆくホームページ作り （size=+0）</FONT><BR>
<FONT size="-1">一歩先ゆくホームページ作り （size=-1）</FONT><BR>
<FONT size="-2">一歩先ゆくホームページ作り （size=-2）</FONT><BR>
<FONT size="-3">一歩先ゆくホームページ作り （size=-3）</FONT><BR>
<FONT size="-4">一歩先ゆくホームページ作り （size=-10）</FONT></P>
</BODY>
</HTML>
```

合は，文字と背景は見る側のブラウザの基本値に従います。たいていは文字が黒，背景が白で表示されます。

## 文字と背景を飾る

今回の講座ではまず，背景を単色で指定する基本タグから伝授しますが，最近は背景に画像を使っているホームページが多く，単色の背景は白または黒が圧倒的になってきているようです。といっても，見やすさという面で考えれば，背景画像の主な色合いと，その上に乗せる文字色のバランスは当然考えなければいけないので，配色という点では同じです。

### ページ内の原則を設定する
<body>

まず，本文タグの最初の部分で，そのページ全体の色の原則を決めましょう。「ここから本文が始まりますよ」という印のタグ，<body>の中には，本文全体の背景色<bgcolor>，文字色<text>，リンク指定をした文字列の色を設定することができます。さらにリンク指定のある文字列には，リンク先を閲覧していない時点での表示色<link>，リンクを選択した時の表示色<alink>，さらにはリンクをすでに閲覧済であることを示す表示色<vlink>の計三つを異なった色で指定することもできます。これを決めておけば，閲覧されていないページをアピールすることもできますし，クリックした瞬間に色を変えてボタンに似た効果をだすこともできます(**第3図・リスト３**)。

これも特に指定しない場合は，ユーザーのブラウザの指定色に一任されます。

また，サイズの大きな背景画像の場合は，低速回線の場合，表示されるまでに時間がかかります。その際，黒っぽい背景画像に白っぽい文字を使用していると，背景が読みこまれるまでの間，文字が全く見えない，なんてことになりますが，これを防ぐためにもあらかじめ見やすい対比となる背景色を設定しておけば，先に読みこまれる背景色が背景画像の表示までの間をカバーしてくれるので，見

る側へのストレスがかなり軽減できます。デキるページにしたい人はぜひこの小技テクニックもぜひ採用してください。

## 一部の文字列の色や大きさを変える
<font color>，<font size>

本文中の一部の文字列だけを変更する方法です。

色を変える場合は，<font color=" ">タグで，"　"の中に背景色で使用したのと同じ16進法を使って指定します。

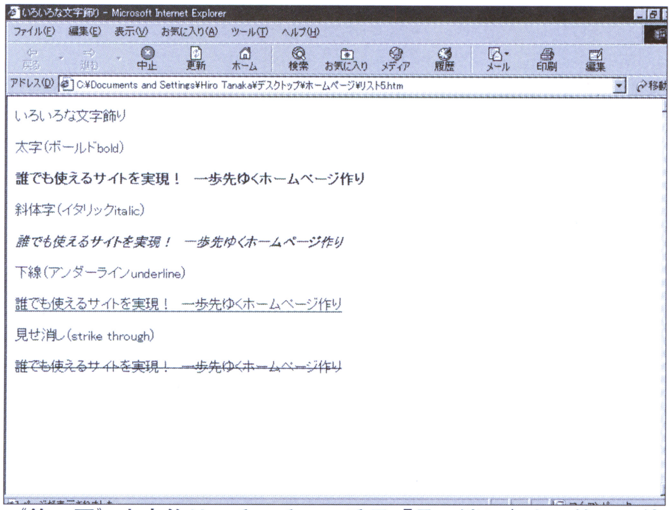
文字の大きさは，<font size="　">タグで変えますが，これには2種類の方法があります。

ブラウザ上で表示できる文字の大きさは全部で7段階ですが，これを1～7の「絶対値」で指定する方法が一つ。そして，ブラウザが基準として指定している文字の大きさにプラス・マイナスいくつと「相対値」で指定する方法が一つです（**第4図・リスト4**）。

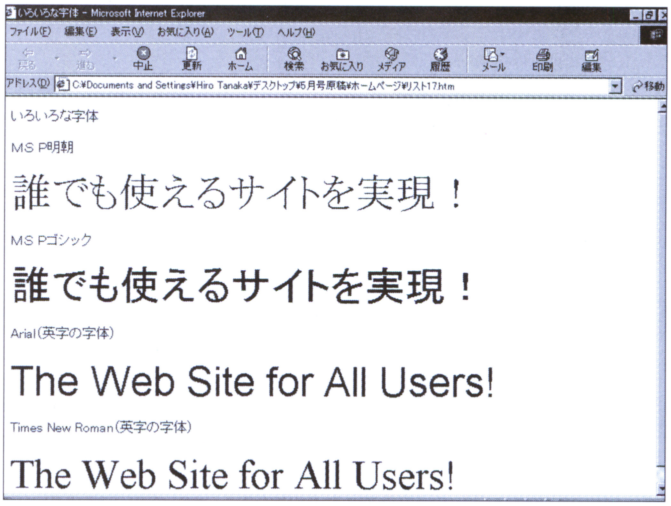
プラス・マイナスの指示は，＋4から0を含めて－2までの計7段階となりますから，＋5や－10という指定をしても，そのとおりには反映されません。あくまでも7段階となります。

とはいえ，文字サイズはユーザーがブラウザで指定する数字によって大きく異なるため，製作者側の指定が反映されにくいところです。インターネットエクスプローラではツールバーの「表示」→「文字のサイズ」で「最小」から「最大」まで5種類の設定が可能ですし，ネットスケープでは「表示」→「テキストサイズ」の項で7種類，「編集」→「設定」→「フォント」で8ピクセルからなんと72ピクセルまでの21段階での設定が可能です。このため，自分のマシン上では問題なく表示されていたレイアウトが，違う設定のブラウザ上ではガタガタになることも珍しくありません。

作る側としては，小さいサイズと大き




《第5図》文字飾りいろいろ。4番目「見せ消し」をお笑いに使う人もいますね。斜体字は英語サイトで「このページの著作権は…」といった断り書きに使うのだとか



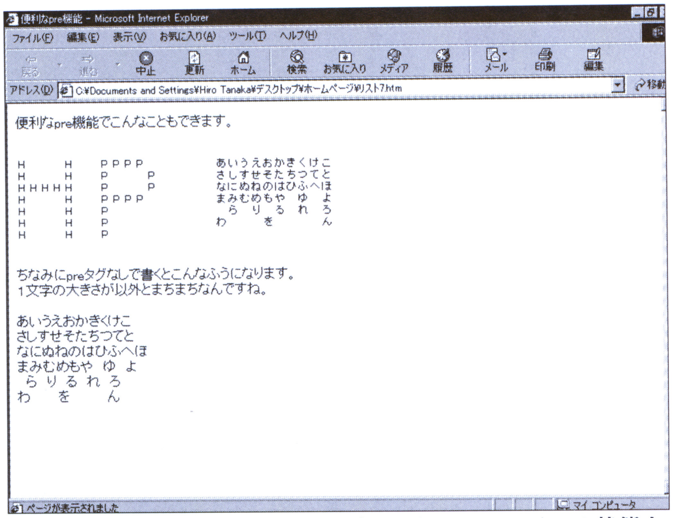

《第6図》字体いろいろ
リストはウィンドウズの搭載字体名で指定しています

《リスト5》第5図のソースリスト

```
<HTML>
<HEAD>
<TITLE>いろいろな文字飾り</TITLE>
</HEAD>
<BODY bgcolor="#ffcc99" text="#000000" link="#ff0000"
vlink="#000099" alink="#0000ff">
いろいろな文字飾り<BR>
<p>
太字（ボールドbold）<BR>
<p>
<b>誰でも使えるサイトを実現！　一歩先ゆくホームページ作り</b><BR>
</p>
斜体字（イタリックitalic）<BR>
<p>
<i>誰でも使えるサイトを実現！　一歩先ゆくホームページ作り</i><BR>
</p>
<p>
下線（アンダーラインunderline）<BR>
<p>
<u>誰でも使えるサイトを実現！　一歩先ゆくホームページ作り</u><BR>
</p>
<p>
見せ消し（strike through）<BR>
<p>
<s>誰でも使えるサイトを実現！　一歩先ゆくホームページ作り</s><BR>
</p>
</BODY>
</HTML>
```

《リスト6》第6図のソースリスト

```
<HTML>
<HEAD>
<TITLE>いろいろな字体</TITLE>
</HEAD>
<BODY bgcolor="#ffcc99" text="#000000" link="#ff0000"
vlink="#000099" alink="#0000ff">
いろいろな字体<BR>
<p>
ＭＳ Ｐ明朝
<p>
<FONT face="ＭＳ Ｐ明朝"size="7">誰でも使えるサイトを実現！
</FONT><BR>
<p>
ＭＳ Ｐゴシック
<p>
<FONT face="ＭＳ Ｐゴシック"size="7">誰でも使えるサイトを実現！
</FONT><BR>
<p>
Arial（英字の字体）
<p>
<FONT face="Arial"size="7">The Web Site for All Users!</FONT><BR>
<p>
Times New Roman（英字の字体）
<p>
<FONT face="Times New Roman"size="7">The Web Site for All
Users!</FONT><BR>
</BODY>
</HTML>
```




**《第７図》pre機能はとっても便利。テキストファイルの状態をそのままブラウザに反映できちゃうスグレ者。ただし多用はブラウザの横幅との関係上，避けた方が無難**

**《リスト7》第7図のソースリスト**

```
<HTML>
<HEAD>
<TITLE>便利なpre機能</TITLE>
</HEAD>
<BODY bgcolor="#ffcc99" text="#000000" link="#ff0000"
vlink="#000099" alink="#0000ff">
便利なpre機能でこんなこともできます。<BR>

<pre>

H      H     P P P P          あいうえおかきくけこ
H      H     P       P        さしすせそたちつてと
H H H H H    P       P        なにぬねのはひふへほ
H      H     P P P P          まみむめもや　ゆ　よ
H      H     P                　ら　り　る　れ　ろ
H      H     P                わ　　を　　　ん
H      H     P

</pre>
<p>
ちなみにpreタグなしで書くとこんなふうになります。<BR>
１文字の大きさが以外とまちまちなんですね。
<p>
あいうえおかきくけこ<BR>
さしすせそたちつてと<BR>
なにぬねのはひふへほ<BR>
まみむめもや　ゆ　よ<BR>
　ら　り　る　れ　ろ<BR>
わ　　を　　　ん<BR>
</BODY>
</HTML>
```

いサイズの両方で表示のテストをしてみて，ある意味，崩れても構わないというくらいのキモチで作るといいと思います。つまり，「こうでなくてはいけない」という思いこみは，文字サイズに関しては捨てようということです。

### さらにいろいろ飾る

<b><i><u><s>

タグを使って文字の見た目を変える方法の代表例です。

太字（ボールドbold），斜体字（イタリックitalic），下線（アンダーラインunderline），見せ消し（strike through）など，おなじみのものが並んでいますね（**第5図・リスト５**）

### 字体を変える

<font face="　">

文字にもいろいろ種類があります。明朝体，ゴシック体，教科書体などの和字の字体の他にも，ArialやTimes New Roman，Courierなど英字の字体もあります。これらを指定するのが<font face>タグです（**第6図・リスト６**）。

ただし，これもユーザー側のマシンに同じ字体が搭載されていないと意味がなくなってしまいます。たとえば隷書体と指定してもユーザー側に隷書体がなければ指定どおりに表示されません。そしてそれは十中八九存在しません。なぜなら字体は，搭載のOSに標準搭載されているものの他は，ユーザーが自発的にダウンロードしない限り存在しないからです。しかも標準搭載の内容はOSごとにまちまちなため，現状ではユニバーサルデザインの大きな障壁になっています。

ここで，マッキントッシュとウィンドウズ双方でなんとか問題なく見られるようにするタグ技をお教えしましょう。

<font face="ＭＳ Ｐゴシック, Osaka">，<"ＭＳ Ｐ明朝, 平成明朝">，<"Tines New Roman, Times">のように，双方の似ている字体をあらかじめ併記するのです。ＭＳ Ｐゴシックとしか指定されていないページをマックユーザーが見た場合は，マシンが「いちばん近い」と判断した字体に表示されます。が，「いちばん近い」といってもなにぶん機械が判断することですから，ちょっとマニアックな字体を指定した場合，人間の見た目にはとんでもない表示がされることもあって，制作者をおおいに驚かすことがあります。

それでも，「タイトルだけはどうしても隷書体を使いたいんだ〜」というような場合には，その文字を画像ファイルにして表示することをオススメします。

## より見やすい・おもしろい構成

今度は，ちょっと技アリ編です。知っている人は分かるけれど知らない人には全然分からない？　そんな機能をもったタグです。

### htmlファイルに打ちこんだ文字列をそのまま表示する

<pre>

htmlファイル上での改行，段落分けはブラウザ上に反映しない，はずなのですが，この<pre>タグを使うと，おっとびっくり反映できてしまうのです。この機能では，行末がガタガタするブラウザの欠点を補うことも可能ですが，文字の大きさは前述したようにユーザーのブラウザ指定でいくらでも変わってしまいますから，ホームページの全体や大部分に使用するのは控えましょう（**第7図・リスト７**）。

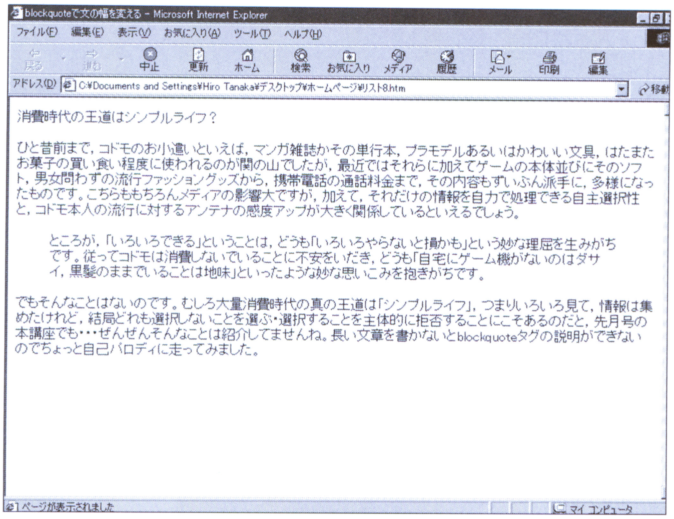
なお，このタグを使った時は，特に</pre>で閉じることをお忘れなく。とんでもない表示になってしまいます。

### 引用文のアタマを一段下げる

<blockquote>

<pre>や改行が使われない限り，ブラウザ上で文字列は左右いっぱいにひろがりますが，このタグを使うと一回り小さい幅で文字列が表示されます（**第8図・リスト８**）。

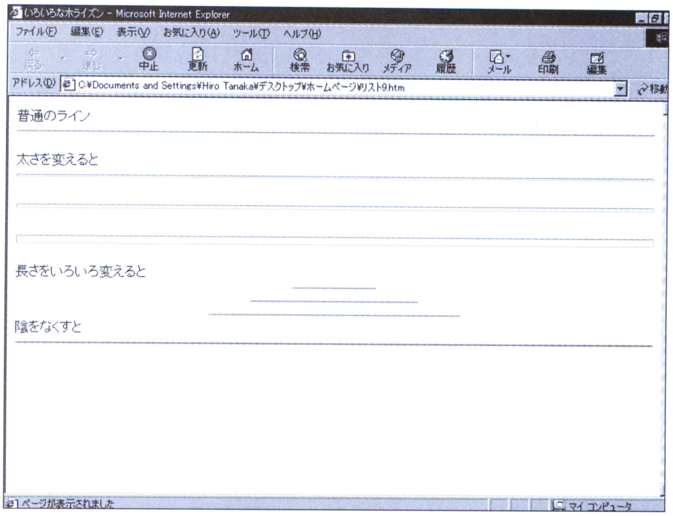
本来の「引用文であることを示す」機能に使ってもよし，一部の文に使用して




《第８図》古くからあるblockqouteタグ。
引用文などを示すのに最適。画面左端に飾りのある背景に使うと字がかぶらないという利点も

《リスト8》第8図のソースリスト

```
<HTML>
<HEAD>
<TITLE>blockquoteで文の幅を変える</TITLE>
</HEAD>
<BODY bgcolor="#ffcc99" text="#000000" link="#ff0000"
vlink="#000099" alink="#0000ff">
消費時代の王道はシンプルライフ？
<p>
ひと昔前まで，コドモのお小遣いといえば，マンガ雑誌かその単行本，プ
ラモデルあるいはかわいい文具，はたまたお菓子の買い食い程度に使われ
るのが関の山でしたが，最近ではそれらに加えてゲームの本体並びにその
ソフト，男女問わずの流行ファッショングッズから，携帯電話の通話料金
まで，その内容もずいぶん派手に，多様になったものです。こちらももち
ろんメディアの影響大ですが，加えて，それだけの情報を自力で処理でき
る自主選択性と，コドモ本人の流行に対するアンテナの感度アップが大き
く関係しているといえるでしょう。
<blockquote>ところが，「いろいろできる」ということは，どうも「いろい
ろやらないと損かも」という妙な理屈を生みがちです。従ってコドモは消
費しないでいることに不安をいだき，どうも「自宅にゲーム機がないのは
ダサイ，黒髪のままでいることは地味」といったような妙な思いこみを抱
きがちです。</blockquote>
でもそんなことはないのです。むしろ大量消費時代の真の王道は「シンプ
ルライフ」，つまりいろいろ見て，情報は集めたけれど，結局どれも選択し
ないことを選ぶ・選択することを主体的に拒否することにこそあるのだと，
先月号の本講座でも・・・ぜんぜんそんなことは紹介してませんね。長い
文章を書かないとblockquoteタグの説明ができないのでちょっと自己パロ
ディに走ってみました。
</BODY>
</HTML>
```




《第9図》ホライズンいろいろ
なんだかホライズンらしくないものもあって面白いです。使い方次第ではクールなレイアウトも可能ですね

《リスト9》第9図のソースリスト

```
<HTML>
<HEAD>
<TITLE>いろいろなホライズン</TITLE>
</HEAD>
<BODY bgcolor="#ffcc99" text="#000000" link="#ff0000"
vlink="#000099" alink="#0000ff">
普通のライン<br>
<hr><br>
太さを変えると<br>
<hr size="3"><br>
<hr size="5"><br>
<hr size="10"><br>

長さをいろいろ変えると<br>
<hr width="100">
<hr width="200">
<hr width="300">

陰をなくすと<br>
<hr noshade>
</BODY>
</HTML>
```

目立たせてもよし，ページ全体に使って左右に余白効果を出しているページもよく見ます。

### ラインいろいろ

<hr>

ページの左右方向に一本の線を引く<hr>タグはよく使われていますが，この線もいろいろ指定することで珍しいものに変えることができます。

長さを変える<size>，太さを変える<width>，陰をなくす<noshade>などを組み合わせただけで**第9図・リスト９**のようなバリエーションが可能です。

## 画像のテキストの配置

画像ファイルも，htmlタグによる表示の世界では「一つの文字」と同じ扱いになります。ですから画像と文字が混在している場合，特にタグ指定がなければ，画像は１行の中に存在する巨大な文字といった扱いで表示されてしまいます。これでは画像が大きい場合にレイアウトが乱れるので，タグ指定をして，より見やすくする必要があります。

### 文字列を画像の上/中央/下につける

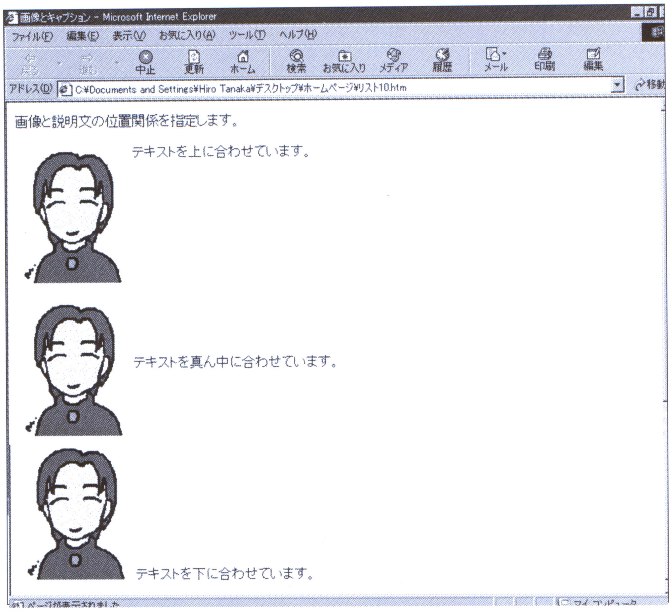
<img src="abc.gif" align="　">

画像につける短い説明文（キャプション）を，画像のどの位置につけるかを指定します。"top"，"middle"，"bottom"の３種があります（**第10図・リスト10**）。

### 複数行の文字列を横に表示する

<img src="abc.gif" align="　">

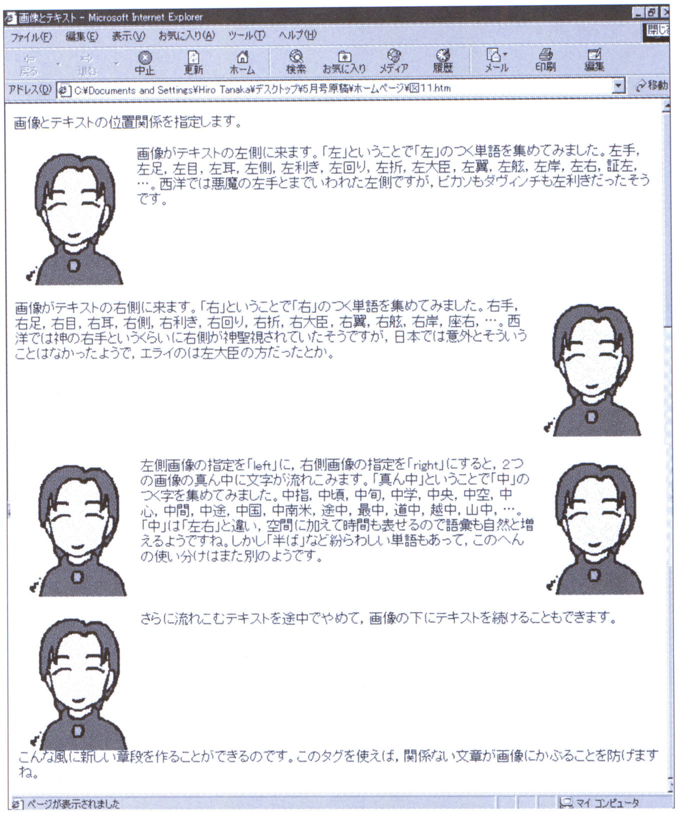
画像の脇を文字列が回りこんでいるような効果を出します。タグはキャプションと同じですが，こちらは絵の右に回るか，あるいは左に回るかを指定するため，"right"，または"left"とします（**第11図上２段・リスト11**）。組み合わせると，**図11**中段のような効果も出せます。




《第10図》画像につける説明文の位置を指定
特に指定がなければ下につきますが，きちんと指定してこそのユニバーサルデザインです

《リスト10》第10図のソースリスト

```
<HTML>
<HEAD>
<TITLE>画像とキャプション</TITLE>
</HEAD>
<BODY bgcolor="#ffcc99" text="#000000" link="#ff0000"
vlink="#000099" alink="#0000ff">
画像と説明文の位置関係を指定します。
<p>
<img src="shiraki.gif" align="top" alt="100万ドルの笑顔">テキスト
を上に合わせています。<p>
<img src="shiraki.gif" align="middle" alt="100万ドルの笑顔">テキス
トを真ん中に合わせています。<p>
<img src="shiraki.gif" align="bottom" alt="100万ドルの笑顔">テキス
トを下に合わせています。
</BODY>
```




《第11図》画像と文章のレイアウトを指定
長い文の中に画像を入れた場合，他のブラウザ環境で見るとガタつきやすいので要注意。しっかり指定しましょう

《リスト11》第11図のソースリスト

```
<HTML>
<HEAD>
<TITLE>画像とテキスト</TITLE>
</HEAD>
<BODY bgcolor="#ffcc99" text="#000000" link="#ff0000"
vlink="#000099" alink="#0000ff">
画像とテキストの位置関係を指定します。
<p>
<p>
<img src="shiraki.gif" align="left" alt="100万ドルの笑顔">
画像がテキストの左側に来ます。「左」ということで「左」のつく単語を集
めてみました。左手，左足，左目，左耳，左側，左利き，左回り，左折，左
大臣，左翼，左舷，左岸，左右，証左，…。西洋では悪魔の左手とまでいわ
れた左側ですが，ピカソもダヴィンチも左利きだったそうです。
<br clear="all">
<p>
<img src="shiraki.gif" align="right" alt="100万ドルの笑顔">
画像がテキストの右側に来ます。「右」ということで「右」のつく単語を集
めてみました。右手，右足，右目，右耳，右側，右利き，右回り，右折，右
大臣，右翼，右舷，右岸，座右，…。西洋では神の右手というくらいに右側
が神聖視されていたそうですが，日本では意外とそういうことはなかったよ
うで，エライのは左大臣の方だったとか。
<br clear="all">
<p>
<IMG src="shiraki.gif" align="left" alt="100万ドルの笑顔">
<IMG src="shiraki.gif" align="right" alt="100万ドルの笑顔">
左側画像の指定を「left」に，右側画像の指定を「right」にすると，2つ
の画像の真ん中に文字が流れこみます。「真ん中」ということで「中」のつ
く字を集めてみました。中指，中頃，中旬，中学，中央，中空，中心，中間，
中途，中国，中南米，途中，最中，道中，越中，山中，…。「中」は「左右」
と違い，空間に加えて時間も表せるので語彙も自然と増えるようですね。し
かし「半ば」など紛らわしい単語もあって，このへんの使い分けはまた別の
ようです。
<br clear="all">
<p>
<img src="shiraki.gif" align="left" alt="100万ドルの笑顔">
さらに流れこむテキストを途中でやめて，画像の下にテキストを続けること
もできます。<br clear="all">
こんな風に新しい章段を作ることができるのです。このタグを使えば，関係
ない文章が画像にかぶることを防げますね。
</BODY>
</HTML>
```

## 横に短い文章を表示し、空白を開けて画像の下に文章を続ける

<br clear="all">

このタグなら画像の説明文だけを脇に表示し，本文は下に続けることができます（第11図下段）。

## 画像と文章の間に隙間を作る

<hspace="　">，<vspace="　">

特に指定しない場合，文字列は画像にぴったり貼りついたようになっていますが，これでは見にくいので隙間を作りたい，という時に使うタグです。

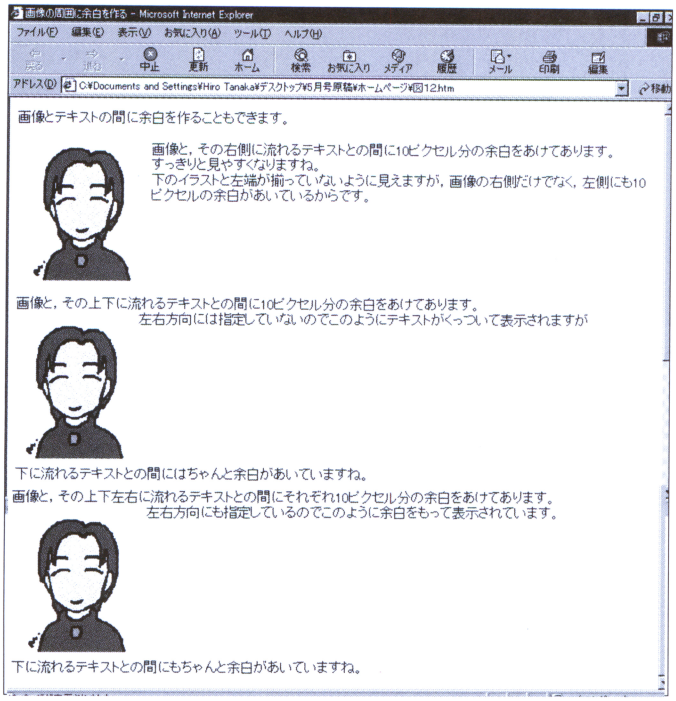
"　"には隙間の大きさをピクセル数で指定します。もちろん左右上下両方を指定することも可能です（第12図・リスト12）。

## こんな技もありますが…

さてここまでで，ホームページ上で表示できるテキストの形をひととおりマスターできました。

ですが，世には「なんじゃこりゃ？」と思わずうなる技アリホームページがまだまだたくさんあります。そのいくつかを種明かししたいと思います。

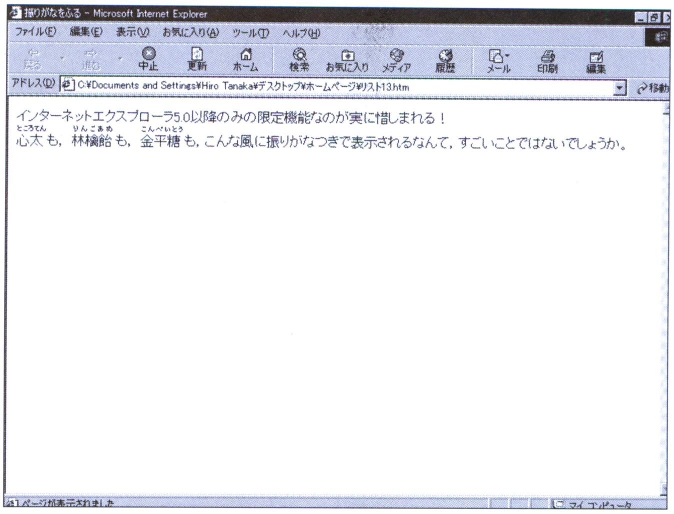
ただし，これらの技のほとんどはユニバーサルデザインには向きません。ブラウザの機能が向上し，いろいろな技が使えるようになって，使えるからにはじゃんじゃん取り入れたいと思うのは確




**《第12図》余白で見やすく。意外と使われていないタグですが，こんなに見やすくなります。真ん中の画像だけ左端がズレてるように見えますが，横方向に余白指定をしていないので画面左端に密着してしまっているためです**

**《リスト12》第12図のソースリスト**

```
<HTML>
<HEAD>
<TITLE>画像の周囲に余白を作る</TITLE>
</HEAD>
<BODY bgcolor="#ffcc99" text="#000000" link="#ff0000"
vlink="#000099" alink="#0000ff">
画像とテキストの間に余白を作ることもできます。
<p>
<p>
<img src="shiraki.gif" align="left" hspace="10" alt="100万ドルの笑
顔">
画像と，その右側に流れるテキストとの間に10ピクセル分の余白をあけて
あります。<br>
すっきりと見やすくなりますね。<br>
下のイラストと左端が揃っていないように見えますが，画像の右側だけでな
く，左側にも10ピクセルの余白があいているからです。
<br clear="all">
<p>
画像と，その上下に流れるテキストとの間に10ピクセル分の余白をあけてあ
ります。<br>
<img src="shiraki.gif" align="left" vspace="10" alt="100万ドルの笑顔
">
左右方向には指定していないのでこのようにテキストがくっついて表示され
ますが<br clear="all">
下に流れるテキストとの間にはちゃんと余白があいていますね。
<br clear="all">
<p>
画像と，その上下左右に流れるテキストとの間にそれぞれ10ピクセル分の余
白をあけてあります。<br>
<IMG src="shiraki.gif" align="left" hspace="10" vspace="10"
alt="100万ドルの笑顔">
左右方向にも指定しているのでこのように余白をもって表示されています。
<br clear="all">
下に流れるテキストとの間にもちゃんと余白があいていますね。
</BODY>
</HTML>
```




**《第13図》これが驚異の振りがな機能！**
**日本人ならゼッタイ欲しい技につき，ユニバーサル化を切実希望の逸品ですね**

**《リスト13》第13図のソースリスト**

```
<HTML>
<HEAD>
<TITLE>振りがなをふる</TITLE>
</HEAD>
<BODY bgcolor="#ffcc99" text="#000000" link="#ff0000"
vlink="#000099" alink="#0000ff">

インターネットエクスプローラ5.0以降のみの限定機能なのが実に惜しまれ
る！<br>
<ruby>
<rb>心太
<rt>ところてん
</ruby>
も，
<ruby>
<rb>林檎飴
<rt>りんごあめ
</ruby>
も，
<ruby>
<rb>金平糖
<rt>こんぺいとう
</ruby>
も，こんな風に振りがなつきで表示されるなんて，すごいことではないでし
ょうか。
</BODY>
</HTML>
```

かに人情ですが，こうした「特化機能」の多くは，別ブラウザでは対応できていないことが多いのです。凝りに凝った結果，他の人には見られないホームページになってしまっては意味がありませんよね。

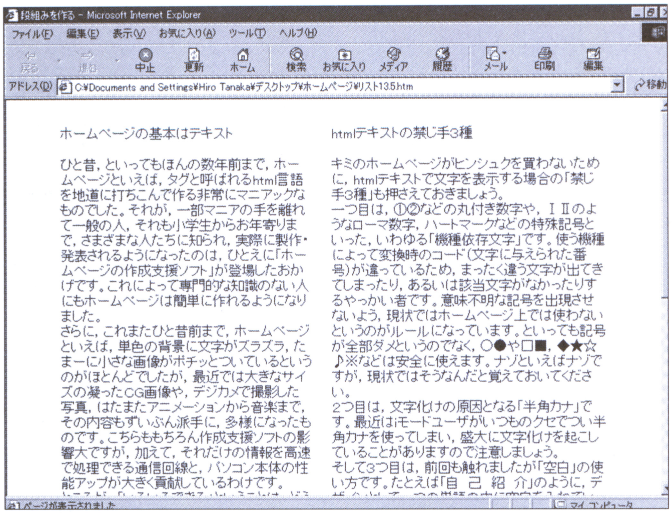
### ルビ（振りがな）を振る

これはインターネット発祥の地アメリカの英語使用者にはまったく関係なく，しかし日本語使用者には切実に必要な機能と言えるでしょう。これがタグで簡単にできてしまうのなら使わない手はない…と思ったらインターネットエクスプローラ5.0以降でのみ動作するとのことで残念。参考までに**第13図・リスト13**を挙げておきます。

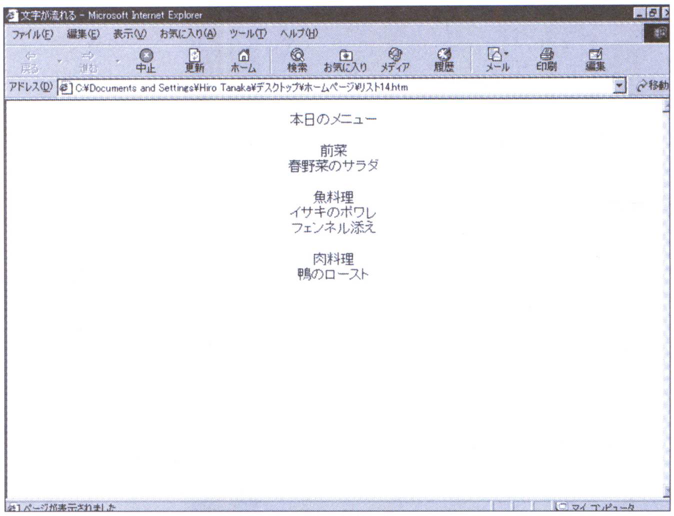
### 新聞みたいな段組のある画面

これはカッコいい…けれども，一つのタグによる指定方法は，残念ながらネットスケープ3.0と4.0のみの機能なのでした（なんと同じネットスケープでも6.0では無視されます！）。ちなみに<multicol cols="2">と書くだけで左右２段組が完成する簡単さでした。

もっともこのレイアウト（**第14図**）は，「テーブル」機能を使えばネットスケープ6.0でもインターネットエクスプローラでも簡単に作成できますのでご安心を。テ




《第14図》これが段組み。これほど便利なタグが最新版では無視！の憂き目に遭うなんて…タグの七不思議です




《第15図》まるで映画のエンドクレジット！
動いているところをお見せできないのが残念ですが，ＩＥ上では下から上へ文字が流れます

ーブルについてはこの講座でも取り上げる予定です。

### 横へ上へ流れる文字
<marquee>

インターネットエクスプローラのみの機能ですが，わりと古くからあるものの一つに<marquee>というタグがあります。

動きには，方向指定の<direction>機能で右から左，あるいは左から右に文字列を流す，また，動きの種類を指定する<behavior>機能として，停まったり<slide>，流れ続けたり<scroll>，左右を行ったり来たり<alternate>させることができます。さらに，4.0からは上下方向への<direction>機能も加わり，映画のエンドクレジットのような動きも可能になりました。もちろん，画面内の文字の位

## コラム　自分のホームページの見え方，みんなはどうやって確かめてるの？　白木ミホ

誰にでも利用しやすいユニバーサルデザインのホームページについて，第１回，第２回と本連載を読んでくれたキミたちの中には，実際にもう自分のホームページを持って，運営している人もかなりいると思います。また，持っていない場合でも，ユニバーサルデザインの世界を知った今，じゃあそういった「違い」はどうやって確かめればいいのか，ホームページのユニバーサルデザイン度チェックとも言えるその方法を知りたい人もいるでしょう。

では，一味ちがう，デキるウェブマスターたちが採用しているチェック方法を一挙公開？しちゃいましょう。現役管理人必見，将来の管理人さんたちもぜひ参考にしてみてください。

1.インターネットエクスプローラとネットスケープの両方でチェックする。コレ基本中の基本ですね。

2.しかもあえて最新版でないブラウザでもチェックする。最新版はあえて使わないというウェブマスターはけっこう多いようです。

3.先月号で紹介した第３のブラウザ・オペラもさっそくダウンロードして見え方チェック済み。ま，単なる新しモノ好きという話もありますが…。

4.それぞれのブラウザで文字サイズを小から大までいろいろ変えて見てみる。あえて文字列のレイアウトが崩れるような大きさにしてみて，それでも見苦しくならないか，内容は理解できるかチェックしましょう。

5.画面から1m離れて見る。近眼の人は眼鏡を外して画面を見てみる。これで色のコントラストがボケなければＯＫ。

6.自分のパソコンと違うOSを搭載しているマシンでチェックする。お父さんのノートパソコン，友達のマッキントッシュ，学校のマシンなどなど，あちこちで見てみる。自分のホームページを宣伝しているのかと思われがちですが，とにかくやるのがコツです。

7.さらにインターネットカフェや電器店の最新製品など，さわれるものならビシバシさわって見まくる。これは結構面白いです…。

8.パームトップ（手のひらサイズ）のモバイル端末などでも見まくる。画面の大きさが根本的に違うといいテストになります。

9.「見え方がおかしい」などといったクレームに対してはお礼の言葉を尽くし，情報を歓迎していることをアピールする。人みな我が師でありますね。

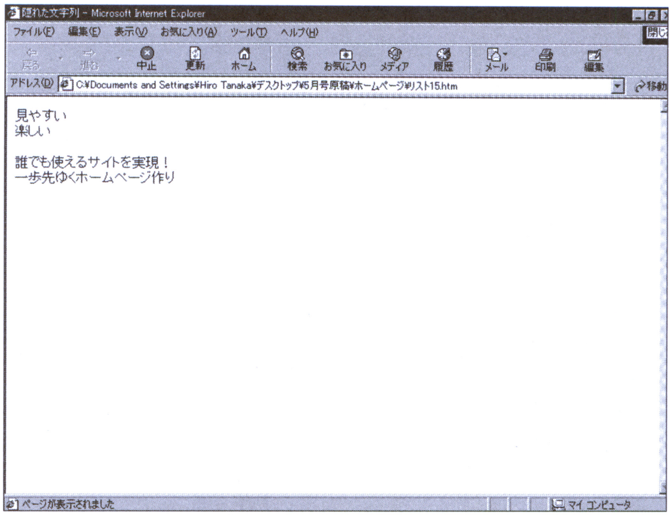
10.他人のホームページに対して，たとえ煙たがられてもクレームを表明する。我こそはユニバーサルデザインの伝道師なり・・・？

一つもクリアしてない人は，この連載を読んで心を入れ替えましょう。三つくらいでしたら，まあそこそこ，フツーの管理人さんです。本連載を読んでさらに努力しましょう。７～８コという人は，ユニバーサルデザインを楽しんでいる姿が目に浮かびます。たいへん素晴らしいことです。

10コ全部やってますという人は・・・プロのウェブクリエイターとなって，明日のネット世界を背負って立って下さい。世界中のネットサーファーがあなたの活躍を待っているはずです。では！

《リスト14》第15図のソースリスト

```
<HTML>
<HEAD>
<TITLE>文字が流れる</TITLE>
</HEAD>
<BODY bgcolor="#ffcc99" text="#000000" link="#ff0000"
vlink="#000099" alink="#0000ff">
<marquee align=top behavior=scroll direction=up loop=1>
<center>
本日のメニュー<br>
<p>
前菜<br>
春野菜のサラダ<br>
<p>
魚料理<br>
イサキのポワレ<br>
フェンネル添え<br>
<p>
肉料理<br>
鴨のロースト<br>
オレンジソースかけ<br>
<p>
デザート<br>
いろいろ<br>
</center>
</marquee>
</BODY>
</HTML>
```




**《第16図》ドロン！隠れ文字**
**テキストファイルにあってブラウザにない，そんな隠れ文字が書ける，制作者御用達のメモタグです**

《リスト15》第16図のソースリスト

```
<HTML>
<HEAD>
<TITLE>隠れた文字列</TITLE>
</HEAD>
<BODY>
見やすい<br>
楽しい<br>
<p>

<!--  ブラウザ上ではこのタグの中の文字列は見えません  ->

誰でも使えるサイトを実現！<br>
一歩先ゆくホームページ作り<br>

</BODY>
</HTML>
```

置<align="　">や，サイズ<hight="　",width="　">の指定もできます。残念ながら誌面ではその肝心の流れがご紹介できません(**第15図**)ので，**リスト14**のみを参考にしてください。

### 番外…ソースにあるのに画面に出ない文字列
<!–　–>

htmlテキストにちょこちょこっとメモをしておきたい，そんな時に重宝するのがこれ。<!–　–>タグに挟まれた文字列はブラウザに反映されません。この機能はユニバーサルなので，大手を振って使うことができます（**第16図・リスト15**）

## まとめ

テキスト一つでもこれだけの機能を駆使することができるホームページの世界は，今後もますます多くの機能を取りこんで広がり続けるはずです。しかし，その広がりが各々のブラウザに特化して孤立独走してしまうのか，それとも互いの機能を取りこみ共存共栄していくのかは，現在こそがその分岐点であるように思われます。ともあれ，世界中の人が見ることのできるもの，それがホームページの最大の特長である以上，この世界に限っては独自の機能が歓迎されないことは確かでしょう。

ところで，ユニバーサルデザインとテキストの関係として，私たちが直面するいちばん大きな問題は，実は私たちの使用する言語が日本語であるということです。世界中の人が見る，という観点から考えると，テキストの究極のユニバーサルデザインは，現時点では英語を使用するということになります。なぜなら，日本以外で発売されるパソコンのOSには日本語の字体が標準搭載されていない，つまり日本語で作成したホームページは日本以外では自発的に日本語フォントをインストールしないと見られないからです。

ホームページと言いながら自国内向けにしか情報を発信しないのだから日本のホームページは閉鎖的だ，と言われても仕方ないのが現状なのです。一時期，日本語ドメインが話題に上りましたが，ホームページにアクセスするためのアドレスにまで日本語の字体が使われたら，海外からのアクセスを著しく制限することになる，というユニバーサルデザインの立場から，これには日本国内からも反対論が出ました。一部の人（日本人）の便利さだけを過剰に追求すると，他の人の不便になるということは，ホームページの世界でも同じなのです。

しかし，行間が少ないといったように，ホームページの仕組みが英語の便利に偏り，他の言語の便利に目を向けていない場合もあるでしょう。それらの点にもユニバーサルデザインの考え方は広がるべきだと私は思います。しかし，そのためにはまず英語で発信しなければならない，というのは矛盾しているようにも見えますが。

次回のテーマは「リンク」です。ではまた。

パソコンの心臓部　マイクロプロセッサ誕生30周年企画

# CPUの歴史と未来をさぐる!

# モバイルプロセッサが ユビキタスな世界を作る!

文・絵／くりひろし

**Dr.D：**ふ〜，やっぱいい天気じゃのう！リゾートじゃなぁ…。

**つぐ美：**ははは，不景気も一段落でほっと一息，ゴールデンウィークは行楽でGO！ですかね。

**Dr.D：**こないだまで年度末あーんどGW進行で忙しかったからのう！久々に羽でものばさんとな。…だからわざわざ南の島まできたというのに…この男ときたら！

**影：**やっぱベーマガ編集部たるもの，どんな時でもモバイル！でしょう！

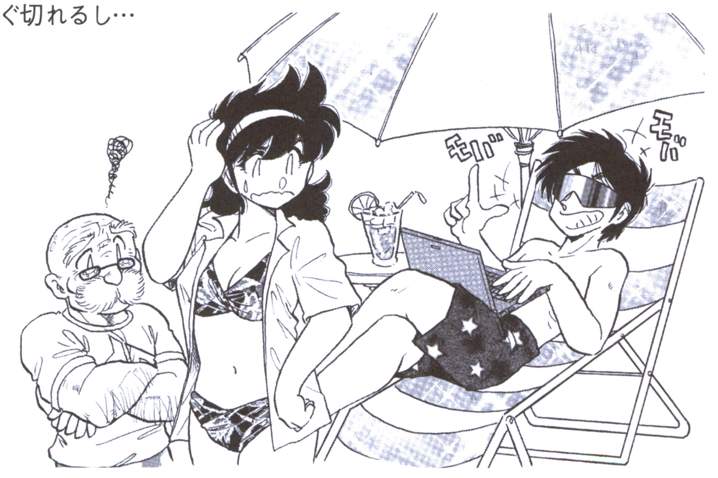
**つぐ美：**あああ，うっとうしいわねぇ，せっかくの休みなのにぃ！だいたいこんなビーチサイドでノートパソコンなんて，すぐにバッテリーが切れるわよ？

**影：**……あ，切れた。がーん，書きかけの記事がぁぁぁ！

**つぐ美：**ほ，ホントにやっとったんかい！…にしても，使い方がヘタだなぁ。バッテリー監視くらい入れなさいよ。

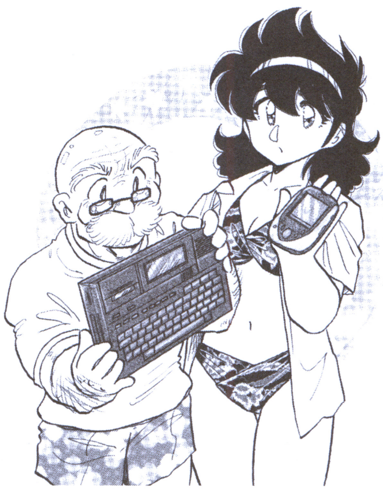
**Dr.D：**まぁ，仕事しとったのは，ん〜まぁ，影さんにしては…偉い…かな？

**影：**ちぇえーっ，や〜れやれ，モバイル機器もいろいろあるけど，一番ダメなのがバッテリーだよなぁ。重いし，すぐ切れるし…



近年のモバイル機器の発達はめざましいものがあるなぁ，と読者のみなさんも感じているでしょう。ｉモード携帯のゲーム機能（ｉアプリ）はもちろん，最近はやりでメジャーＰＣメーカーからも新製品そろい踏みのPocketPC，インターネット連携を始めたPDA……これに，この春からは無線LANによるホットスポットサービスも始まり，いよいよ，いつでもどこでもコンピュータ製品を持ち歩いて，ネットワークに接続して好きなことができる――これをユビキタス・コンピューティングといいます――時代が到来してきました。

インターネットの普及は世界を変えましたし，携帯電話の普及は生活スタイルを変えました。ブロードバンド＋常時接続はインターネットの世界をさらに進化させ，そして無線ＬＡＮとＩＭＴ－2000（FOMA）が，ついにユビキタスな世界を作ろうとしています。そこで，肝心なのが端末となるモバイル機器，その中心となるのがモバイルプロセッサです。しかし，モバイル機器の性能はもちろん，携帯性能としてのバッテリーの持ちなど意外なところで足かせとなり，一方でマルチメディアへの対応に欠かせない高機能を実現しているのもまたモバイルプロセッサなのですが…。



## まずはちょっとモバイルパソコンのいま・むかし

**Dr.D：**今時はよくモバイル，モバイルっていうけど，かつては持ち運べるパソコンのことをモバイルＰＣとは呼ばなかったな。

**つぐ美：**ノートＰＣ？今でも言いますよ。

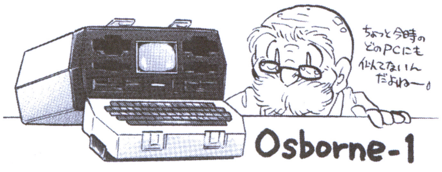
**Dr.D：**それは形状のことじゃよ。モバイル(Mobile)というのは移動できる，とか機動性がある，という意味なのは知っておると思うが，携帯電話などを使って，旅先などでパソコン通信をできるようになった頃からモバイルって呼んでいるなぁ。

**影：**持ち運べるって意味なら，「ハンドヘルド(Hand Held)」ですよね！欧米ではノートＰＣのことをいまでもハンドヘルドＰＣって言いますね。

**Dr.D：**お，よく知っておるな。そいつはあくまで「手で」持ち運べる，という意味でな，その前は「ポータブル(Portable)」PCって呼ばれておったのだ。

パソコンの元祖は1975年のAltair 8800ですが，同じ年に世界初のプログラム電卓であるＨＰ－65がヒューレットパッカード社より発表されています。プロセッサは４ビットでしたが，近代コンピュータの基本要素の一つである，「プログラムを組める」という機能だけを考えれば，Altair8800と同じ年に実現していたのは驚異といえるでしょう。ただ，あくまでも電卓ですから計算式以外のプログラムはできません。現在のちょっと高級な計算式の記憶機能のある電卓程度だったのです。

さて，世界で最初にポータブルパソコンを「自称」したのはコモドールのPET-2001ですが…って，シャレはさておいて，本当に世界初のポータブルパソコンといえるのは1981年のオズボーン-1です。現代のどのコンピュータにも似ていなく，どちらかといえばＬＡＮアナライザなどの計測機器に似ているのですが…重量13ｋｇ，閉めた時にはハンドルになるフタが，開けばそのままキーボードになるデザインは，まさしくモバイルコンピューティングを意識していました。…さすがにバッテリーはついていないので，あくまでも移動先での使用に限定されていました。



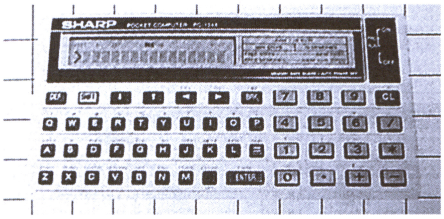

一方，ポケットコンピュータは現在のＰＤＡの祖先といえるでしょう。1980年に発表されたシャープのPC-1211は，プロセッサこそ４ビットでしたが，なんとＢＡＳＩＣが搭載されており，気軽にプログラミングを楽しめたのでした。いわゆるプログラム電卓との最大の違いはＢＡＳＩＣのような高級言語が使えるかどうかで，後発のカシオPB-100なんかもプログラム電卓と名乗っていましたが，周辺機器の充実ぶりや性能的にも完全にポケットコンピュータといえるものでした。

電池で動くので，本当の意味でモバイルコンピューティングが20年前から実現できていたのです。本誌にも長い間ポケットコンピュータのゲームプログラムの投稿がよく掲載されました。１～２，３行しか表示されないＬＣＤディスプレイで，アイディアの限りを尽くしたゲームが多く作られました。

グラフィック機能はないに等しい（１文字分６×８ドット単位でキャラクタデータを生成することは出来ましたが…）ので，数字や文字，記号を組み合わせてキャラクタを表現しました。現在のインターネットで流行っているＡＡ（アスキーアート）の源流がここにあります。しかもそれを動かしてアニメーションまでさせていたのですから，人間何も無けりゃ無いでどうにかしておもしろくしていくものです。

その後，本格的に現在のノートＰＣの元祖といえるのが，1987年の米国版T-1000（ターミネーターとは関係…無い…よね？）から始まる東芝Ｊ-3100，ダイナブックです。デザイン的には1983年のシャープＰＣ-5000がクラムシェル型（貝殻が開くようにLCDが起きあがって操作出来るようになるのでこの名がついた）を先に発表していますが，T-1000あたりになると開くディスプレイ（モノクロですがプラズマディスプレイ！）がずっと大きく，VGAフルサイズ表示可能（カラーではありませんが…）になっており，周辺機器の内蔵や拡張ポートの付き方などは現在のノートＰＣとほとんど変わりません。

何より，かつてアラン・Ｃ・ケイ（パソコンの父と呼ばれる）が1968年に提唱した「ダイナブック」という未来のコンピュータの理想像を名前に持ってきたため，Ｊ-3100をはじめとした当時の「ラップトップ（ひざの上）」コンピュータは大ヒットしたのでした。当時搭載されていたプロセッサは，別にモバイル用



▲1980年登場のシャープ「PC-1211」の後継機「PC-1245」。ポケットコンピュータはPDAの先祖！？

◀1982年登場のエプソン「HC20」は元祖A４ハンドヘルドPCだ

**初期モバイルＰＣの歴史**

| 年 | メーカー | 機種名 | CPU | 特徴 |
|---|---|---|---|---|
| 1975 | HP | HP-65 | 不明,4bit | ＰＣの祖先，「プログラム電卓」 |
| 1977 | コモドール | PET-2001 | MOS6502 2.5MHz | 自称「ポータブル」PC（マジ？） |
| 1980 | シャープ | PC-1211 | SC61860 | BASIC搭載のポケットコンピュータ |
| 1981 | オズボーン | Osborne-1 | Z80A 4MHz | 重量13kg,世界初の持ち運び可能PC |
| 1982 | エプソン | HC20 | 6301 600KHz×2 | 元祖Ａ４サイズハンドヘルドＰＣ |
| | TANDY | TRS-80 m100 | HC20に同じ | HC20のOEM，米国で大ヒット |
| | NEC | PC-2001 | μPD7907 6MHz | HC20に並ぶ元祖A4ハンドヘルド |
| | SANYO | PHC-8000 | NSC-800 ?MHz | 小型で軽量 |
| | カシオ | PB-100 | 不明,4bit | カシオ初のBASIC搭載ポケコン |
| | シャープ | PC-1251 | SC61860 | ポケコンのベストセラー |
| | カシオ | FX-702P | 不明,4bit | 倍速改造ができたプログラム電卓 |
| 1983 | NEC京セラ | PC8201 | 80c85 2.4MHz | なんでもついてて多機能 |
| | 松下 | JR800 | 63A01 4.9MHz | B５小型ハンドヘルドの走り |
| | シャープ | PC-5000 | 8086 3MHz | 元祖クラムシェル型デザイン |
| 1984 | エプソン | HC80 | Z80 2.4MHz | サブCPUに6301 600KHz |
| | エプソン | HC40 | Z80 3.7MHz | サブCPUに7508 270KHz |
| 1987 | 東芝 | T-1000 | 80c86 10MHz | J-3100のオリジナル，米国版 |
| 1988 | NEC | UltraLite | 80c86 10MHz | 米国版，B５ノートＰＣの元祖 |
| 1989 | 東芝 | J-3100 | 80c86 10MHz | 実質的な国内Ａ４ノートPCの元祖 |
| | アップル | MacPortable | MC68000 | Macintosh初の携帯型 |

などということはなく，ふつうの8086系プロセッサでした（さすがに省電力に適したCMOS版を使っていますが）。

**つぐ美：** ええっと，昔のラップトップPCやノートPCやなんかって，結局モバイル用のプロセッサは使ってなかったのかしら？
**Dr.D：** うむ，専用のプロセッサは基本的になかった。そもそも昔の8086とかってサイズも小さくて，モバイルにそのまま流用しても問題なかったんじゃな。まぁ，搭載の都合上，パッケージ形状を変えたり，基板に直付け（ソケットを使うと厚くなるから）することはあったが，中身は同じものだ。
**つぐ美：** …そういや，どう考えてもノートＰＣに入らないなって思えたのはPentiumⅡのSlot1対応PPGAパッケージとかよね。あのでっかいカセット型。
**Dr.D：** うむ，たしかにPentiumⅡからじゃないかな，まがりなりにもモバイルの名称がついたのは。最初は単なるパッケージサイズの小型化から入ったんだがな。

## モバイルプロセッサは発熱対策と共に

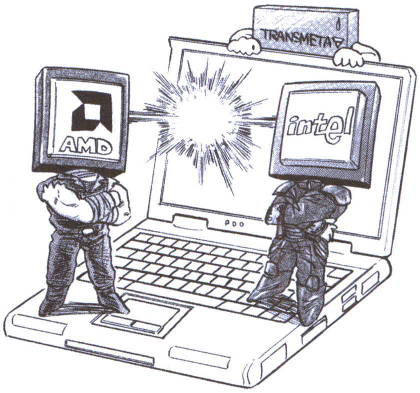
**Dr.D：** で，だ。モバイルプロセッサというと，冷却問題は切っても切れない問題なのだが…ＣＰＵが熱くなったらどうすれば良いか？
**つぐ美：** そりゃ，冷やしますよねぇ？ヒートシンクとかでっかいし…そうそう，昔プロセッサのクロックアップ改造時に流行った，ペルチェ素子を使うというのは？小さいですよ，あれは。
**Dr.D：** あー，あれは片面冷えるんじゃが，反対の面がその逆数倍熱くなるんじゃよ。しかも，ペルチェを動作させるのにそもそも電力がいるから，結局貴重なバッテリーを食って，しかも全体では逆に熱が増えるという…
**つぐ美：** がーん。しろうとにはおすすめできない…
**影：** 冷やすってのは，なんかアタリマエすぎるなぁ？熱くならないようにするんだよねぇ？…使わない，ってのはどうかな？
**Dr.D：** お，影くんにしてはやるなぁ，

**Intel モバイルプロセッサの熱設計電力の変化**

| 時期 | プロセッサ | 動作クロック | ＴＤＰ |
|---|---|---|---|
| 2002.9?～ | Celeron(Northwood) | 1.4GHz? | 30W? |
| 2002. 3～ | Mobile Pentium4-M | 1.4GHz～2.0GHz? | 25W～30W? |
| 2001.12～ | Celeron(Tualatin) | 1.06GHz～1.3GHz | 22W |
| 2001. 6～ | Mobile PentiumⅢ-M | 866MHz～1.26GHz | 22W |
| | Mobile PentiumⅢ-M LV | 733MHz～1.0GHz | 12W |
| | Mobile PentiumⅢ-M ULV | 700MHz～933MHz | 7W |
| 2000.12～ | Celeron(Coppermine) | 700MHz～933MHz | 19W |
| 2000. 7～ | PentiumⅢ | 800MHz～1GHz | 19W |
| | PentiumⅢ LV | 600MHz～750MHz | 12W |
| | PentiumⅢ ULV | 500MHz～600MHz | 7W |
| (参考) | Pentium4 (Desktop) | 2.0GHz | 76W |

TDP：熱設計電力 Thermal Design Power　LV　：低電圧版 Low Voltage
ULV：超低電圧版 Ultra Low Voltage

でたらめなようで核心をついた答えだ！
**つぐ美：** なななな，なによ，それが答え？
**Dr.D：** 正しくは，「使ってないぶんは使わない」，じゃな。



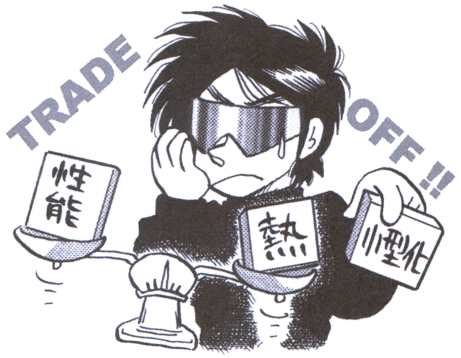

近年のPentium4採用のノートPCに見られるような高速性は，そもそもモバイルプロセッサには求められていませんでした。パソコンメーカーもそれほど力を入れる気はなかったはずなのですが，ＡＭＤ対Intelのプロセッサ対決の余波がモバイルプロセッサにまで及び，メーカー側も「あるんだったら使った方が売れるだろう」ということになったわけです。

本来のモバイルの基本はあくまでも携帯性であり，長時間駆動，小型，軽量が要求仕様です。そのためには電池・バッテリーの持ちが良い様に，消費電力は極力低く，すなわち動作クロックもそれほど高くないことになります。これが変わってきたのが，デスクトッププロセッサのモバイル化，ノートＰＣの普及によるプロセッサ戦争の戦場がモバイルに移動してきたこと，NTTのFOMA(IMT-2000)のように，携帯端末の広帯域（ブロードバンド）化，インターネット連携サービスに伴う高性能への要求が増えていることです。

高速化への要求の代償は発熱です。発熱こそは「バッテリーの持ち」と並ぶノートＰＣの２大問題の一つです。もちろん，電気製品である以上，発熱はさけられないものではありますが，発熱がすぎると熱暴走などの誤動作を起こしたり，場合によってはプロセッサが焼けて壊れてしまいます。では，どうやって発熱を抑えれば良いのでしょう。発熱対策にはいくつか方法があります。

１）では，大型のヒートシンクほど

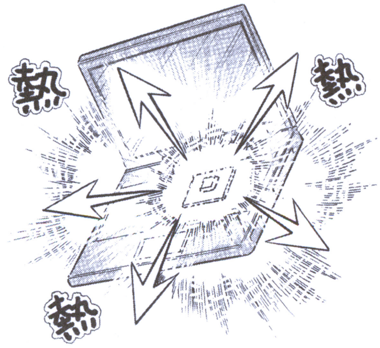
１）熱を逃がす冷却装置を取り付ける　→　ヒートシンク／冷却ファン
２）発熱しにくい作りにする　→　製造プロセスを小さくする
３）動作クロックを下げる＋動作電圧を下げる




冷却効果は高いのですが，ノートパソコンなどでは大きさに限界があります。冷却ファンもプロセッサに直付けにはできませんが，基板全体を冷やすことで冷却します（したがって，ノートPCをクッションやカーペットの上などで使うのは冷却上あまり奨められません。なるべく熱伝導性の高いスチールの机なんかが本当は良いのです）。最近，不凍液によって液晶パネルもふくめて全体を冷却する，いわゆる液冷システムも発表されています。また，プロセッサの表面に製造の過程で熱伝導を高める部品，ヒートスプレッダ（IHS：Integrated Heat Spreader）が最初から装着されるものもあります。

2）は前回でも説明した，プロセッサの構造を変えていくものです。発熱現象そのものはプロセッサの回路部分を通り抜ける電気抵抗のためです。数ミリ四方の集積チップの中に，数千万個のトランジスタが生成されているマイクロプロセッサでは，プロセッサを構成するトランジスタとその回路に流れる電流の量を少なくすることで，発熱量を下げることができます。回路の線，間隔が細く，密度が小さいほど発熱量＝消費電力を減らすことができるため，プロセッサをどれだけ細かく製造できるか，すなわち製造プロセスを小さくすることです。一世代前の0.18μm製造プロセスから最新の0.13μmに下げることで，クロック性能を50％向上させることができるのです。

3）は，動作クロックと動作電圧が一定の比例関係にあるということです。動作電圧を上げれば動作クロックを向上させることができるのですが，その分発熱も比例して上がってしまうわけです。したがって，この方法は性能とのトレード・オフになってしまうのです。もちろん，プロセッサの構造上，動作電圧を小さくできれば,それにこしたことはありません。




## クルーソーが来るそうだ！

**Dr.D：**ながらくライバル関係にあったAthlonとPentiumだが，ここにきて新たなライバル，Crusoeががんばっておるな。

**つぐ美：**クルーソー？ロビンソン・クルーソーが元ネタですかね？

**Dr.D：**うーん，ま，新興プロセッサの冒険といえばいえるので，そうかも？

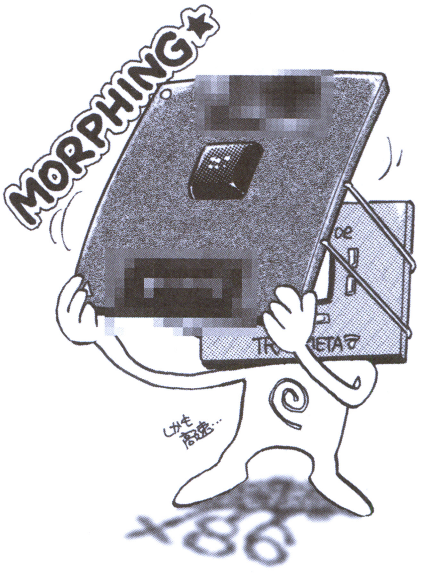
**影：**いやいや，やっぱクルーソー警部でしょう！不可能を可能にする！

**Dr.D：**ピンクパンサーかね…いくらなんでもそれは…。

**つぐ美：**それをいうなら，Crusoe自体がピンクパンサーよね。最強の宝石，プロセッサですよ。

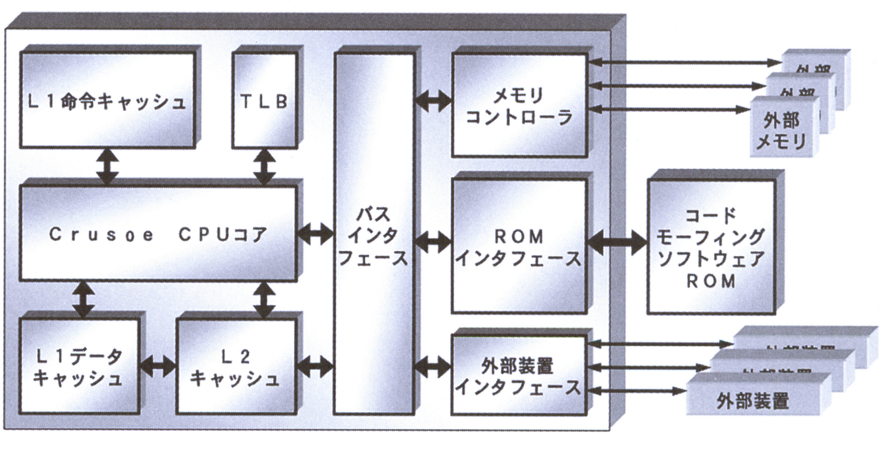
**Dr.D：**うまいのぅ，つぐ美くんは！

1999年，モバイルプロセッサの戦場に第三のメーカーが現れました。それが米国のプロセッサメーカー，




Transmeta社です。同社はLinuxで有名なリーナス・トーパルス氏も参加しているITベンチャーの超有望株です。同社の製品であるCrusoe(クルーソー)は，AMD社のAthlon，Duronなどと同じくIntel社のいわゆるx86プロセッサ(80386～Pentium4，80x86)用のソフトウェアが動作します。

しかし，Athlonなどのようにx86プロセッサ命令がそのまま動作する，x86互換プロセッサではありません。Crusoeが画期的だったのは，コードモーフィング（命令変換）技術によってx86命令をCrusoe用のシンプルなRISC命令に変換して（そう，Crusoeは基本的にRISCプロセッサなのです）高速に実行することなのです！　モーフィングとは変換とか変身と訳される言葉ですが，Crusoeの場合まさしく，単なる変換というよりは変身と呼べるほどです。そ




▲Crusoeプロセッサの基本構造

の，コードモーフィングは以下の手順で行われます。

1）まず，入ってきたｘ８６命令をatomコード（RISC系のシンプル命令）に変換します。

↓

2）次に，冗長な（余分な）命令を削除したりまとめたりして，最適化されたatomコード群にします。

↓

3）複数の最適化atomコードをグループ化し，Molecule（分子）コード，すなわちCrosoeにとってのマシン語命令であるＶＬＩＷ（超長命令語：Very Long Instruction Word）にします。

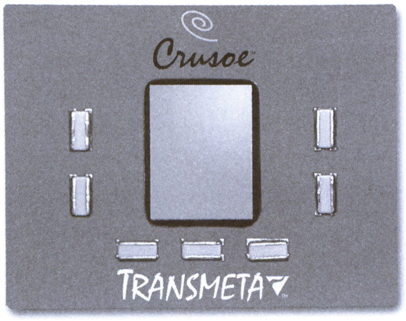
↓

4）Moleculeコードをパイプライン処理によって高速に実行します。

↓

5）モーフィングされた命令はキャッシュ（一時蓄積）され，同じ命令は再モーフィングの必要なく，より高速に効率よく実行することができる。

Crusoeにはコードモーフィングの他にも強力な武器があります。それは強力な省電力機能です。Crusoeの省電力技術，LongRunは動作中に，周りの状況を判断しながら，CPU電圧と動作クロックを切り替えるものです。それだけ聞くとIntel社のSpeedStepと似ていますが，SpeedStepは決まった電圧とクロックの組み合わせしかできませんが，LongRunでは段階的にこきざみに変えていくことで，フルパワー状態から10％程度の超省電力状態までを最適な電圧，クロックにできるのです。



▲Transmeta社の「Crusoe」TM5600

そもそもがCrusoeはRISC+コードモーフィングによって，トランジスタ集積度もPentiumⅢの1/4以下です。そのため，最新のCrusoe TM5800は，1GHzという高クロックでありながら，最大のTDP熱設計電力が７Ｗと，Mobile PentiumⅢ ULVモデル 700MHzと同等の省電力を達成しています。

Crusoeはその高性能と省電力のおかげで，いわゆるＢ５ミニノートサイズＰＣや，さらに小さいコンパクトサイズのＰＣに多く採用されています。大規模な国際展示会で受賞するほどの評判になり，採用機種も多く出てきています。ただし，x86コードのためのモーフィング技術が，本来のRISCならではの実行速度を使えなくしている点で，期待していたほどPDAでの採用は無いようです。

Intelも次々と超低消費電力(ULV)型のモバイルプロセッサを出していますが，さすがに性能と省電力のトレードオフがCrusoeほどうまく行かず，次世代のモバイルプロセッサ，コードネーム"Banias"に期待をかけたいところです。

**これまでに出たCrusoeプロセッサのシリーズ**

| 年 | 型番 | 動作クロック | 特徴 |
|---|---|---|---|
| 2001 | TM5800 | 1GHz～ / TM5500 800MHz | 最新，ついに1GHzを突破！ |
| 2001 | TM3400 | 600MHz | 第二世代のローエンド向け |
| 2000 | TM5600 | 700MHz / TM5400 667MHz | ハイエンド？向けCrusoe |
| 2000 | TM3300 | 500MHz / TM3200 400MHz | 話題になった初代Crusoe |

## そういえば，RISCってなんだっけ

**つぐ美：**あの，さっきからRISCだから速いのどーのと言ってますが。

**Dr.D：**む，ちゃんと説明したこと無かったかな。

**影：**リスクって，危険を承知でっていう…。

**Dr.D：**いや，そういうんではなくて。

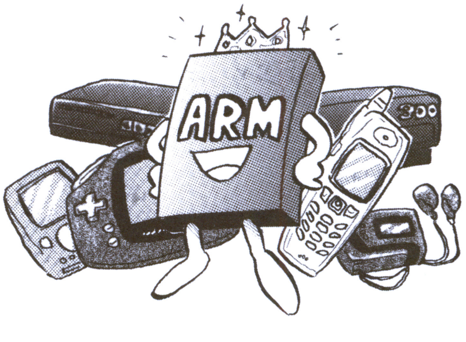
**影：**あ，わかった，残り物のパンを揚げて砂糖をまぶした…。

**つぐ美：**それはラスクだぁぁぁ！

RISC(Reduced Instruction Set Computer)プロセッサはなぜ速いのでしょうか？RISCはその名のとおりに，命令セット（プロセッサを動作させるマシン語命令の一つ一つ）を縮小しています。元々は1970年代のマイクロプロセッサ初期の頃からあったもので，簡略化された命令をパイプライン処理で複数を並列に処理することで高速化を図るものです。たとえば，通常のx86系プロセッサでは，マルチメディア支援機能やらなにやら，多くの複雑な命令セットを持っていますが，実際にプロセッサが動作するときには一部の命令ばかり使って，ほとんどの命令は使われていないのです。

その一方で，普段は使われない命令を実行するためには多くの複雑な処理部が必要になり，よけいな時間がかかるようになっていました。そのため，多機能高機能であるにもかかわらず，トランジスタの集積度や動作クロックの上昇ほどには比例して処理速度が上がらないジレンマが発生していたのです。そこでRISCプロセッサでは，よく使う命令を残し，あまり使われない命令を思い切って捨て，よく使う命令の組み合わせで処理するようにしました。その上で，命令をパイプライン処理することで大幅な高速化をはかったものです。

RISCプロセッサは，元来業務用／技術計算用のワークステーションで使われていました（Sun社のSPARCプロセッサや，DEC社のAlphaチップなど）。パソコンではPowerPCがRISC技術を使



っていますから，動作クロックに比べると遙かに高性能なのはRISCの効果でしょう。そして，今回話題のCrusoeやARM(以降で出てきます)など，シンプルで省電力なのにハイ・パフォーマンスを求めるプロセッサでは，RISCが有利なようです。

## PDAの熱い戦い！

**つぐ美：** そもそも，ノートパソコンなんてカッコ悪いわ。大きいし，重いし，バッテリーすぐに切れるし。
**影：** ど，どうしろと？
**つぐ美：** やっぱ PDA（Personal Digital Assistant）でしょう！小さい，軽い，バッテリー長持ち，最近はインターネットもできるしね！
**Dr.D：** わしだって持ってるぞ。
**つぐ美：** むむ，Palmですね！それはそれでクールなんだけど，今一番ホットなのはやっぱりPocketPCですよね！
**Dr.D：** うーむ，確かに今までPDAを出していなかったパソコンメーカーが軒並み参入してきておるなぁ。案外2002年最大のヒットになるのかも？
**つぐ美：** ここでもIntelのXScaleが台風の目ですよね！AMDは出さないのかなぁ？
**Dr.D：** む，何か勘違いをしておるな。同じＡではじまるのでも，XScaleのライバルはARMなのだ。
**つぐ美：** ２文字まで合ってるのに～。
**Dr.D：** た，確かに…なんかあるのかのう？そ，それよりPalm系，DragonBallはドウヨ？
**影：** ほえ，ドラゴンボール？…って，あの，漫画にあった？あれ，好きだったんだよなぁ。なんか新作描いてくんないかな？つーか，なんでドラゴンボール？ひょっとして一星球とか入っているのか？
**つぐ美：** ほ，ほんとにそうならちょっとすごいんだけど…。
**Dr.D：** ま，商標上は分野が違えば同じ名前がついてもいいからな。まぁ，ドラゴンボールは欧米でも大人気じゃったから，神龍にでもあやかりたかったんじゃろ。
**つぐ美：** 最近はXScaleに押されそうになってますけど，かつてはPDAといえばDragonBallって時期はありましたねぇ。
**影：** うーむ，国産はザウルスがあるんだけどなぁ。竜つながりで仲間にしてくんない？
**つぐ美：** ちょっと違うと思うなぁ～

### モトローラDragonBall系ＣＰＵ

| 型式 | 愛称 | 周波数 | 特徴 |
|---|---|---|---|
| MC68328 | DragonBall | 16MHz | 初代DragonBall。68000の直系。 |
| MC68EZ328 | DragonBall EZ | 20MHz | 周辺機器コントローラを内蔵。 |
| MC68VZ328 | DragonBall VZ | 33MHz | CPUコアを新型FLX68000に変更。 |
| MC68SZ328 | DragonBall Super VZ | 66MHz | 高速化，周辺機能を集約。68系最終？ |
| MC9328MX1 | DragonBall MX1 | 200MHz | CPUコアはARM920T。ついに非68000系。 |

◀初代DragonBallのMC68328

◀IBMの「WorkPad c505」は，DragonBall VZ 33MHz搭載

PDAはもともと紙の手帳でやっていたようなスケジュールや住所録のような情報管理ができる（PIM:Personal InformationManagement機能）コンピュータ応用製品のことです。したがって，Palm系のように本来のコンセプトに近い製品が長らく主流でしたが，今やインターネット，携帯電話ネットワークとの連携が求められており，WindowsCE系のPocketPCとの競合になってきました。

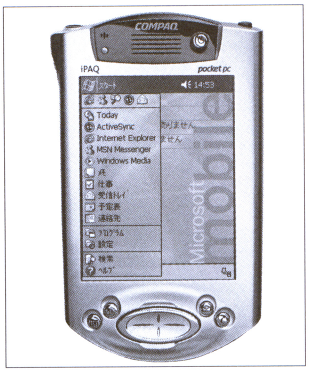
DragonBallはMotorola社が1995年に発表したPDA向けのプロセッサです。CPUの中枢部はあの68000系で，型番号的にもMC68328と，MC68300のバリエーションであることを物語っています。DragonBall(MC68328)はその消費電力の低さと，なにより68000系であるため価格がこなれていたため，ノートＰＣよりもずっと安価でなければならなかったPDA市場では歓迎されたのでした。

DragonBallは，PDAブームを巻き起こしたPalmに採用されたことから，急速に広まり，元祖Palm以外にもHandSpringのVisor，SONYのCLIE，IBMのWorkPadなど多くの機種に採用されています。

### ＡＲＭ系ＣＰＵ

| メーカ | 愛称 | 型式 | 周波数 | 特徴 |
|---|---|---|---|---|
| ARM | ARM7 Thumb | ARM7TDMI | 88/133MHz | PDA，MP3プレーヤ |
| | | ARM720T | 75/100MHz | デジカメ，ゲーム機 |
| ARM | ARM9 Thumb | ARM920T | 200/250MHz | PDA，携帯，デジカメ |
| | | ARM940T | 185MHz | PDA，携帯，デジカメ |
| ARM | ARM9E Thumb | ARM926EJ-S | 180～250MHz | PDA，携帯，デジカメ |
| | | ARM966E-S | 190～250MHz | PDA，携帯，デジカメ |
| ARM | ARM10 Thumb | ARM1020/1022 | 325MHz | 次世代PDA |
| Intel | StrongARM | SA-1100/1110 | 100/200MHz | Intel製ARM |
| Intel | XScale | PXA250 | 200/300/400MHz | XScaleアーキ |
| Intel | XScale | PXA210 | 133/200MHz | XScaleアーキ |

▲ARM系のIntel XScale PXA250



▲COMPAQの「iPAQ Pocket PC H3800シリーズ」には，ARM系のIntel StrongARM SA-1110 206MHz搭載

その後，DragonBall Super VZ (なんつー名前でしょう(^^;))で66MHzかつ通信機能などの周辺機能を集約した決定版にたどりつきましたが，強敵があらわれました。それが，影のモバイルプロセッサの帝王，ＡＲＭです。ARM社(Acorn RISC Machine)は 英国のAcorn Computerを母体としていますが，1990年にAPPLE社の出資などで設立された，MotorolaやIntelに比べるとずっと若い会社です。元々は教育用コンピュータ向けに安くて省電力のエコノミックなプロセッサを研究していたのですが，この点に目をつけたのが同じ頃に需要が急速に高まったモバイル業界，インターネット業界です。

ARMはローコスト，低消費電力，そして小型であることから，携帯電話(携帯電話も立派なコンピュータ応用製品ですね）やインターネットのルーターなどに多く採用されました。特に，欧州やアジアで勢力を誇るGSM方式や，北米中心のCDMA方式の携帯電話ではほとんどがARMを採用しており，その出荷台数はそこらのノートＰＣやＰＤＡなど足下にも及ばない数になっているのです（昨年2001年の生産数は５億個を超えると言われています…ちなみにDragonBallシリーズは２千万個くらい)。

ＡＲＭは，身近なところでは何を隠そう，ゲームボーイアドバンスにも使われているそうです（公式ではありませんが，ARM720Tと言われています)。スーパーファミコンを凌ぐ描画性能，電池の持ちを考えると，ARMシリーズの高性能ぶりと省電力性能の高さがうかがえるでしょう。

ARM社は,実は自前でプロセッサを製造しておらず，ARMプロセッサの設計を行い，それをプロセッサメーカーが製造しています。たとえば，日本ではエプソン，海外では世界有数の半導体メーカに成長したSamsung，そしてあのIntel社も生産しています。特に，IntelのARMは改良型の"StrongARM"を生産しており（元々はDigital社が生産していたのを買収した…)，昨年市場が立ち上がり始めた，ＰＤＡ用の新しいOS，PocketPC向けのプロセッサとして，2001年に発表されたコンパックのiPAQや東芝のGENIOから採用され始めました。

ちなみにPocketPCは，2000年にMicrosoft社がWindowsCE 3.0をベースに開発した，第４のジャンルＰＤＡ市場を担うＯＳです。ターゲットは明らかに対抗するPalm系PDAのOSであるPalm OSを狙っています。WindowsCEで開発された技術が生かせるため，元々Windows系のソフトも開発していたパソコン系のメーカーでの採用が目立ちます。最新バージョンとしてPocketPC2002がすでに発表されており，インターネット通信機能，セキュリティ対応などが強化されています。

Motorola社は，ARMのRISCプロセッサとしての高性能，Intel製StrongARMのPDA市場での普及に驚異を感じ，昨年ついにARMからの技術を導入することにしました。それが最新プロセッサである"DragonBall MX1"です。一世代前ですが，十分に高性能のARM920Tをベースに開発しており，DragonBallシリーズの速度，省電力性能はさらにアップすることになりました。Palmも，今年発表したPalm OS 5.0からARMをベースにしたプロセッサへの対応を行っています。旧DragonBall系（=68000系）はエミュレーションや互換ツールで対応するそうなので，実質的に長年続いたMC68000の系譜はとぎれることになります。ちょっと寂しいですね。

さて，MotorolaがARM技術を取り入れている間に，一方のIntel社はStrongARMをベースに自社の技術を入れて，超省電力機能をもった強力なプロセッサ"XScale"を開発しています。処理性能のアップには並列処理を高速化できるパイプライン機能，省電力には動作状況に応じて動作クロックや電圧を下げるSpeed Step機能などを盛り込まれたXScaleは，現在最高速度の"XScale"PXA250 400MHzで，実に１ボルト以下の電圧で動作します。DragonBall MX1が200MHzで約３ボルトなので，今年中に出る予定の次世代ARM技術を取り入れたMX2に期待したいものです。

なお，国産独自PDAの雄シャープのザウルスは，初期は独自16ビットプロセッサ，後に日立のRISCプロセッサであるSuperH RISC Engineシリーズ（たとえばパワーザウルスではSH-3 45MHz）を採用していました。ちなみに上位プロセッサのSH-4はセガのドリームキャスト，PERSONAやブロードバンドルータにも採用されている高速プロセッサです。おまけに言うと下位のSH-2はセガ・サターンでデュアルＣＰＵとして使われていた，本誌の読者には結構なじみの深いプロセッサでした。…で，最新のザウルスM1-Exシリーズでは，今やStrongARM 206MHzを採用していたのでした。もともとRISCプロセッサ前提の構造なので問題は無いのですが，Palm系の高速処理やマルチメディア処理への対応を考えると，当然のなりゆきだったのかもしれません（だからといってＯＳはPocket PCにはならない)。

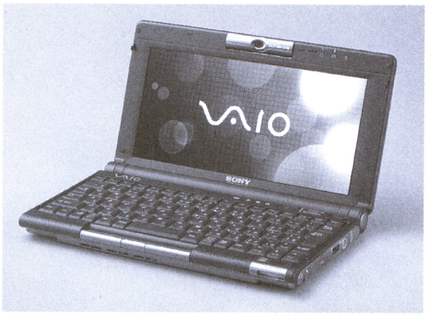
## 次世代モバイルプロセッサよりも,ひょっとしたら大事なもの

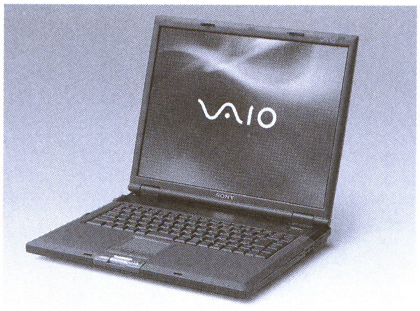
**つぐ美：** それにしても，次世代のモバイルプロセッサってどうなるのかしら？

**Dr.D：** うーん，基本的にはデスクトッ

### Intel vs AMD モバイルプロセッサ1ＧＨｚ突破の状況

| | |
|---|---|
| 2000.3.6 | 参考：AMD デスクトップ用 Athlon 1GHz |
| 2000.3.8 | 参考：Intel デスクトップ用 PentiumIII 1GHz |
| 2001. 5 | Mobile Athlon 4 1GHz　ついにモバイルも1GHzを突破 |
| 2001. 7 | Mobile Pentium Ⅲ-M 1.13GHz |
| 2001.12 | Mobile Duron 1.1GHz |
| 2002. 1 | Mobile Athlon 4 1500+ |
| 2002. 2 | Mobile Celeron 1.13GHz |
| (以降は予測) | |
| 2002.夏 | Mobile Pentium4-M 2GHz？ |
| 2002.秋 | Mobile Athlon 4 2000+？ |
| 2003.春 | Intel Banias 1GHz or 1.5GHz？ |






▲SONYのノートPCも，mobile Athlon4，mobile Pentium4，mobile PentiumⅢ，mobile Celeron，Crusoeなどバリエーションもさまざま。写真はPCG-C1MR/BP（Crusoe TM5600 667MHz）＝左と，PCG-GRX90/P（mobile Pentium4 1.70GHz-M）＝右

プ向けのプロセッサからの技術のスピンオフが中心なのは間違いないが…。Intel社は2003年の前半にも"Banias"という現在のPentium4 Mobileとは違う省電力アーキテクチャの製品を出してくるようだ。Pentium4以上の性能で，ＴＤＰは２０Ｗクラス！

**つぐ美：** え，それはすごいかも…正直，１ＧＨｚを突破したあたりから性能はともかく，ハイエンドのプロセッサっておっきなフルサイズのノートＰＣにしか採用されなくって,がっくししてたのよね。Ｂ５ミニノートなんて相変わらず１ＧＨｚは遠い世界で…

**つぐ美：** ま，水冷ノートパソコンもいいけど，結局重さが犠牲になるから，最初っから省電力性能の高い方がありがたいわよね。

**Dr.D：** そういう意味では最も大事な技術は何かというと，バッテリーの持ちなのだよ。バッテリーさえ軽くて大容量であれば，プロセッサの省電力機能なんて現状程度でもかまわないわけだ。正直，水冷にでも頼らなければ,これ以上の冷却はできないし，発熱に応じて，出来る範囲内ではもう十分省電力できておる。もはや物理的な限界が見えてきているのだ。

**つぐ美：** そうか，それで最近「燃料電池」によるノートパソコンのバッテリーが研究されているんですね。

**Dr.D：** うむ，もう試作レベルではほとんど出来ておる！重さ半分で，一度の燃料供給でバッテリーの持ちは５倍とも言われている。今年の暮れか，来年の初めにも製品が出てくると期待されているな。問題は燃料の補給，安全性だ。

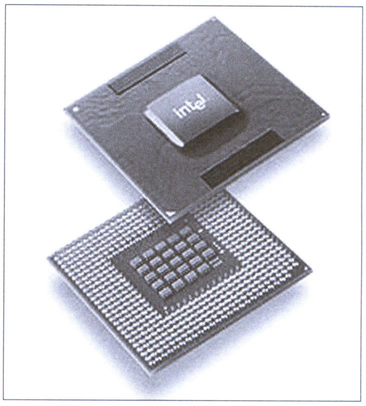
**つぐ美：** そうか，アルコールだったりするから，引火の危険とか，どこで買えるか，とかが大事になってきますね。ううん，技術よりも流通の問題か。

**Dr.D：** まずは今年の夏を乗りきらんとなぁ。初夏のビーチサイドはそれなりじゃが，今年は桜も記録破りに早かったし，夏は猛暑！をワシは予感しとるよ。

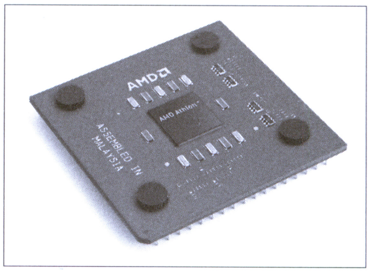
**つぐ美：** そうね，景気回復ももうちょいなんで，オフィスのエアコンは節約モードかしら？暑そうねぇ…

**影：** へへん，心頭滅却すれば火もまた涼し，なのだ！パソコンは熱暴走してナンボやぁぁぁ！燃える男の！熱いモバイル！

**Dr.D：** なんかワケわからなくなっとるが，ま，パソ破じゃないからいいか…




◀mobile Pentium4—M




▶AMDのmobile Athlon4

BM基礎講座

# ベーシックプログラミング教室

## 第2回 分岐処理と繰り返し


**講師紹介**

**川上朝幸**（Kawakami Tomoyuki）
1971年8月31日　大阪府生まれ

音楽系専門学校卒業後 93年カプコン入社。サウンドエンジニアとして，大ヒット作"ストリートファイター２"，"ヴァンパイア"，"ロックマン"シリーズをはじめ"バイオハザード"，"ディノクライシス"など，多くのサウンド制作を経て00年退職，翌01年"スタジオスターノーツ"を起業。サウンドエンジニアとしての活動を軸にプログラミング，パソコンスクール講師，ホームページ制作，ビデオ制作，アマチュア・インディーズバンドの支援など幅広い活動を行っている。

STARNOTES
http://www.starnotes.jp
basic@starnotes.jp


この講座では，Windows上で実行することができ，誰でも容易に入手することができるフリーソフトを利用します。この講座はBASICとして基本的なN88BASICに基づいた内容ですのであらゆるベーシックで応用することができます。

☆フリーで登録されているベーシックプログラミング・ソフトをいくつか紹介します。

●N88互換BASIC for Windows95
<http://www.vector.co.jp/soft/win95/prog/se055956.html>

●ActiveBasic
<http://www.vector.co.jp/soft/win95/prog/se117283.html>

講座内ではこれらを互換BASICと呼ぶことにします。

みなさんこんにちは。今年もやってきました恒例の花粉症…某野球ゲームの育成モードじゃないですけど，花粉症はホントに体力とやる気を消耗させてくれます。医者嫌いなので病院には行かず，なんとか自分で「体質改善！」と叫びながら苦しんでる毎日です。本誌ベーマガが発売されるころには，少しはマシになっていればいいんですが…。

さて，ベーシックプログラミング教室 第1回はいかがだったでしょうか？今月もシンプルな形でわかりやすく進めていきたいと思います。前回と違い，第2回はサンプルプログラムもたくさんでてきます。リストを見ながらがんばってプログラムの入力に慣れていってください。最初は慣れない単語（命令語）がたくさんでてきてややこしいですが，慣れてくれば速く正確に入力できるようになってきます。また単純なバグも出なくなってきます。

僕がプログラムを始めたころは，ファミリーベーシックで延々とベーマガに掲載されているプログラムを入力していました。今のキー入力のスピードは，きっとそのころ身についたんだと思います。PRINTとかINPUTみたいな命令語はもう手が完全に覚えてしまっています。

いろいろなプログラムを入力することも，プログラムをマスターするために重要な事のひとつです。さあ，明日のプログラマーを目指してみなさんがんばっていきましょう！

### プログラムを入力しよう！ その1

それでは早速，互換BASICを起動してサンプルプログラム04を入力してみましょう。プログラムの入力は基本的に半角英数文字で行います。REM(')を使ったコメント文や文字列にのみ全角文字を使用することができます。

プログラムを入力したら早速実行してみましょう。

実行画面に，

＊サンプルプログラム04

```
10 REM プログラムの分かれ道
20 CLS
30 PRINT "1000円持っています。"
40 INPUT "缶ジュース(¥120)を何本買いますか";A
50 B = 120 ＊ A
60 IF B > 1000 THEN 90
70 PRINT"合計";B;"円で，おつりは";1000-B;"円です。"
80 END
90 PRINT "お金が足りません。":GOTO 40
```

**1000円持っています。**
**缶ジュース(¥120)を何本買いますか？**

と表示されるので好きな数字を入力します。

ここでは，9を入力してみます。キーボードから9を入力した後 [Enter] を押してください。

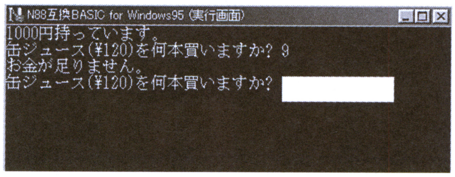
プログラムが次に進み,

**お金が足りません。**

と表示され,

**缶ジュース(¥120)を何本買いましたか?**

と再び表示されます。

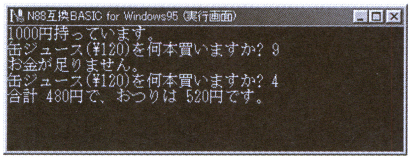



**▲サンプルプログラム04の実行画面**

次に, 4を入力してみましょう。すると,

**合計 480円で, おつりは 520円です。**

と表示されプログラムは終了します。




**▲サンプルプログラム04の実行画面**

## プログラム解説

先月号のサンプルプログラム01に少し手を加えたプログラムです。このプログラムは60行目に分かれ道があります。**IF ～ THEN**が, その命令語です。

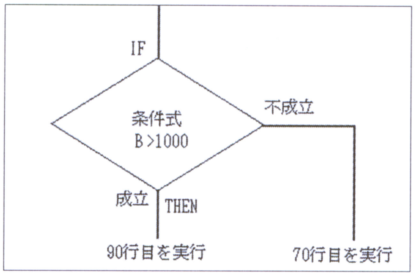
**○IF 条件式 THEN A**

“もし(=IF) 条件式が成立するならば (=THEN) Aを実行する。” という命令です。Aには, 命令語もしくは行番号が入ります。条件式が成立しない場合そのまま次の行を実行します。

それでは60行目を見てみましょう。

**60 IF B > 1000 THEN 90**

この命令は, “もしBが 1000よりも大きいならば, 90行目にジャンプする。”という命令です。Bは, 50行目の**B=120＊A**という式から, 缶ジュースを買った本数の合計金額になるので, 合計金額が1000円よりも多ければ90行目にジャンプ, 合計金額が1000よりも小さければ次の70行目が実行されます。このようにIF ～ THEN命令を使うとプログラムに分かれ道を作ることができます。




条件式では, 次の記号を使うことができます (カッコ内は一般数式記号)。

| | | |
|---|---|---|
| = | | 等しい (=) |
| <> | >< | 等しくない (≠) |
| < | | 右辺よりも小さい (<) |
| > | | 右辺よりも大きい (>) |
| <= | =< | 右辺と同じか小さい(右辺以下≦) |
| >= | => | 右辺と同じか大きい(右辺以上≧) |

90行目にある**GOTO 行番号** というのは “指定した行番号にジャンプする。” という新しい命令語です。プログラムの流れを強制的に変えることができます。

**PRINT**命令で画面に「お金が足りません。」と表示した後に, **GOTO**命令で40行目にジャンプしています。

**IF～THEN**命令には, もうひとつ使い方があります。

**○IF 条件式 THEN A ELSE B**

これは “もし(=IF) 条件式が成立するならば (=THEN) Aを実行し, 条件式が成立しないならば (=ELSE) Bを実行する。”という命令です。A, Bには命令語もしくは行番号が入ります。また一つだけでなく複数の命令語を: (コロン) でつないで書くこともできます。

**IF A=1 THEN PRINT "ＯＫ！":B=2:GOTO 100 ELSE 200**

“もしA=1 ならば, 画面に「ＯＫ！」と表示し, Bに2 を代入。そして100行目にジャンプ。A=1でない時は200行目にジャンプする” という命令になります。

**【ワンポイント】～条件式**

条件式では, 複数の条件式を**AND**や**OR**を使って組み合わせることができます。

**・IF 条件式1 AND条件式 2 THEN A**
条件式1かつ条件式2。二つの条件式が同時に成り立つ場合Aを実行します。

例) IF I>1 AND I<10 THEN A
Iが1よりも大きくかつ10よりも小さい場合Aを実行します。

**・IF 条件式1 OR 条件式2 THEN A**
条件式1または条件式2のどちらかが成り立つ場合Aを実行します。

例) IF I=1 OR J=1 THEN A
I=1 またはJ=1 のときAを実行します。

**IF ～ THEN ～ ELSE**命令は, プログラムに欠かすことのできない, とても大事な命令です。これからもどんどん出てくるので, しっかり覚えておきましょう！

## プログラムを入力しよう！その2

次のプログラムに出てくる新しい命令語も, プログラムには欠かすことのできない命令です。早速サンプルプログラム05を入力してみましょう！

＊サンプルプログラム05

```
10 CLS
20 FOR I=0 TO 7
30   PRINT "ベーマガ最高！"
40 NEXT I
50 END
```

プログラムを入力したら早速実行してみましょう。

実行画面に,「ベーマガ最高！」の文字が8行表示されます。

**▲サンプルプログラム05の実行画面**

これまでに学んだ命令語でプログラ

ムを書いた場合，

```
10 CLS
20 PRINT "ベーマガ最高"
30 PRINT "ベーマガ最高"
40 PRINT "ベーマガ最高"
50 PRINT "ベーマガ最高"
60 PRINT "ベーマガ最高"
70 PRINT "ベーマガ最高"
80 PRINT "ベーマガ最高"
90 PRINT "ベーマガ最高"
100 END
```

と，同じ命令の繰り返しで，やたらと長いプログラムになってしまいます。8行ならなんとかなっても，20行30行と必要になった場合，とても効率が悪い感じがしますね。サンプルプログラム05を見ると，わずか数行で同じ結果になっています。ここでは便利な繰り返し命令について説明します。

## プログラム解説

ここで登場する命令語は**FOR 〜 NEXT**命令です。

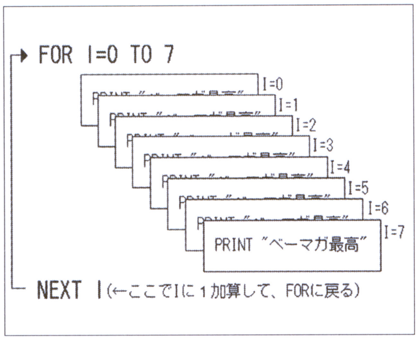
**○FOR 変数=初期値 TO 終値 STEP 増加値**
**〜繰り返す命令〜**
**NEXT 変数**

これは，まず変数に初期値を代入して，**FOR**と**NEXT**間の命令を実行します。**NEXT**まで実行すると変数に増加値を加え再び**FOR**に戻ります。このとき変数の値が終値を超えていなければ，再び**FOR**と**NEXT**の間の命令を繰り返します。

増加値が1のときは**STEP**を省略することができます。

実際にサンプルプログラム05がどういう風に動いているか見ていきましょう。

20行目 **FOR I=0 TO 7** は，変数Iに0を代入し変数Iが7を超えるまで，**FOR**と **NEXT I** の間にある命令語 **PRINT "ベーマガ最高！"**を実行します。




**FOR 〜 NEXT** の中にさらに**FOR 〜 NEXT** を入れることもできます。これを入れ子（ネスト）といいます。次のプログラムを入力してみましょう。

＊サンプルプログラム06

```
10 CLS
20 FOR I=1 TO 5
30 FOR J=1 TO 5
40 PRINT I;
50 NEXT J
60 PRINT
70 NEXT I
80 END
```

実行すると，1〜5の数字が画面に表示されます。

▲プログラムの実行画面

20行目の**FOR** はIが5を超えるまで **NEXT I** との間にある命令を繰り返し実行します。

30行目の**FOR**はJが5を超えるまで **NEXT J** との間にある命令を繰り返し実行します。

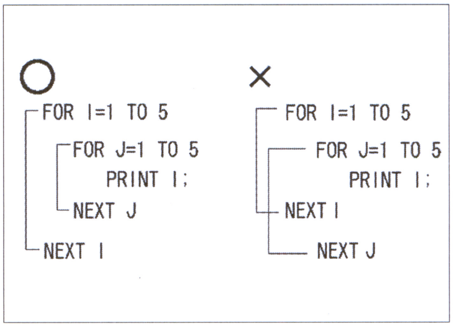
20行目の**FOR I 〜 NEXT I**が1回目の繰り返しを行った(I=1)とき，30行目の**FOR J 〜 NEXT J** はJが1〜5になるまで5回命令を実行することになります。**FOR I 〜 NEXT I**が2回目の繰り返しを行った(I=2)とき，**FOR J 〜 NEXT J** はJが1〜5になるまで再び5回命令を実行することになります。

**FOR I 〜 NEXT I**はIが5になるまで繰り返されるので，**FOR J 〜 NEXT J** の中にある**PRINT I;** 命令は5回×5回の計25回実行されることになります。

40行目の**PRINT** 命令は変数Iの後に;（セミコロン）がついているので，表示後改行されません。30行目と50行目の**FOR 〜 NEXT** が実行されている間は同じ行にIの値が表示されます。60行目で**PRINT** 命令が単独で使用されています。このときに改行の処理がされ次から一行下に表示されることになります。

70行目でIに1が加算されI=2となります。2行目の表示がすべて2になっていますね。

**FOR 〜 NEXT** を入れ子にするときは，下の図のように正しく設定しなければいけません。




## サンプルプログラムを入力しよう！

**IF 〜 THEN** と **FOR 〜 NEXT** の二つの命令語を覚えたことで，プログラムの幅がとても広がりました。いくつかのサンプルプログラムを紹介しますので入力し実行してみてください。

またサンプルプログラムの中で新しい命令語も出てきます。簡単な命令ですが，とても使用頻度の高いものなので同時にマスターしてください。

**【ワンポイント】〜インデント**

30行目の**PRINT**命令の前に半角スペースをいくつか入れています。**FOR 〜 NEXT** の繰り返しの中にある命令はこのように行を下げておくと，プログラムが見やすくなります。

```
20 FOR I=0 TO 7
30 PRINT  "ベーマガ最高"
40 NEXT I
```

↓半角スペースを入れる。

```
20 FOR I=0 TO 7
30   PRINT "ベーマガ最高"
40 NEXT I
```

＊サンプルプログラム07

```
10 '九九の表示
20 CLS
30 FOR I=1 TO 9
40   FOR J=1 TO 9
50     LOCATE J＊3 , I
60     PRINT I＊J
70   NEXT J
80 NEXT I
90 END
```

九九を表示するプログラムです。

**FOR〜NEXT**の入れ子になっていて，I=1のときJ=1,2,3,…,9 と繰り返され，60行目で１の段の計算が行われます。I=2のときも同様に繰り返され2の段の計算が行われます。

新しい命令語として50行目に**LOCATE**命令が登場しています。

**LOCATE X座標,Y座標**

この命令はPRINT命令で表示する位置を設定するものです。X座標とY座標の値は半角文字単位になっています。

サンプルプログラム07では，X座標をJ＊3として，Jが大きくなるたびに3ずつX座標を増やし，Iが大きくなるたびに行を変更しています。

＊サンプルプログラム08

```
10 CLS
20 X=1    'X座標
30 Y=10   'Y座標
40 W=500  'ウェイト
50 FOR X=1 TO 20
60   LOCATE X,Y
70   PRINT " A";
80   FOR I=0 TO W:NEXT I
90 NEXT X
100 END
```

アルファベットの“A”が右に移動するプログラムです。

**FOR 〜 NEXT**命令でX座標を変化させ**LOCATE**命令で“A”を表示する位置を移動させています。ポイントは70行目の**PRINT**命令で，“A”の左に半角のスペースが入っています。このスペースが重要な役割をしています。試しにスペースを消して実行してみてください。

80行目の**FOR 〜 NEXT**はプログラムの実行を遅らせるためのウェイト処理を行っています。

次の命令に行くまでの時間稼ぎをしているわけです。40行目のWの値を変えることで文字の移動速度が変化します。

### 【ワンポイント】〜FOR〜NEXTとGOTO

FOR 〜 NEXT で繰り返される処理の中に，GOTO命令などでジャンプすることはできないので注意しましょう。逆にFOR 〜 NEXT からジャンプすることはできます。

```
20 FOR I=0 TO 7
30   PRINT "ベーマガ最高"
40 NEXT I
50 GOTO 30
```

30行目にジャンプすることはできません。

```
20 FOR I=0 TO 7
30   IF I=5 THEN PRINT "ベーマガ最高":GOTO 50
40 NEXT I
50 PRINT I
```

FOR〜NEXTからジャンプで抜けだすことはできます。

＊サンプルプログラム09

```
10 CLS
20 PRINT "モンスターが現れた！"
30 INPUT "1=戦う，2=防御，3=逃げる";A
40 IF A=1 OR A=2 THEN 100
50 IF A=3 THEN 120
60 PRINT
70 GOTO 20
100 PRINT"モンスターは強かった・・・GAME OVER"
110 END
120 PRINT "無事逃げきることができた! Happy END"
130 END
```

瞬間的に終わってしまうゲーム（？）です。**IF〜THEN**命令でストーリの分岐点を作ることができるのでアドベンチャーゲームを作ることができます。40行目で複数の条件式を使っています。

## 練習問題にチャレンジ

今回も新しい命令語がたくさんでてきましたね。それでは，練習問題にチャレンジしてみましょう！

**問１．** 次のプログラムを実行すると（ア　　　　）と表示される。

```
10 CLS
20 A=5
30 IF A=4 THEN PRINT "合格 “ ELSE
PRINT "不合格"
40 END
```

**問２．** 次のプログラムを実行すると（イ　　　　）と表示される。

```
10 CLS
20 FOR I=0 TO 5
30   IF I = 3 THEN PRINT I
40 NEXT I
50 END
```

**問３．** 次のプログラムの空白を埋める。

```
10 CLS
20 FOR X=0 TO 30
30   FOR Y=0 TO 20
40     PRINT X ＊ Y;
50   NEXT （ウ　　　　）
60 NEXT （エ　　　　）
70 END
```

**問４．** 次のプログラムを実行すると（オ　　　　）と表示される。

```
10 CLS
20 A=3
30 FOR I=0 TO 50
40   IF A=3 AND I=10 THEN 80
50   IF B=2 OR I=100 THEN 90
60 NEXT I
70 END
80 PRINT “失敗” :END
90 PRINT “成功” :END
```

## 次回予告

繰り返し処理として使われるもうひとつの命令（WHILE〜WEND）と，サブルーチン，配列など次回も盛り沢山の内容でお届けします。

（練習問題 解答）

| | | |
|---|---|---|
| ア→不合格 | イ→3 | ウ→Y |
| エ→X | オ→失敗 | |

この講座で使用しているプログラムデータは下記のホームページでダウンロードすることができます。
http://www.beeplus.or.jp/bm/basic/
☆BM基礎講座「ベーシックプログラミング教室」
制作 スタジオスターノーツ
http://www.starnotes.jp/
協力 有限会社 ビープラス
http://www.beeplus.or.jp/

# ゲームを作ってみよう！

**舞台は，星音(ほしね)高校パソコン研究部。４人の部員たちが初めてのゲーム制作にチャレンジ！**
**彼らも「ベーシックプログラミング教室」を読みゲームを作ろう！！と意欲満々！！**
**第2回となる今回，果たしてゲーム制作はどれくらい進んでいるでしょうか！？**

**登場人物**

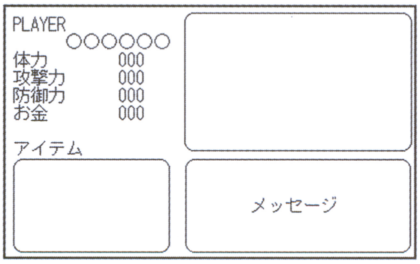
「ちぃさん」(ちぃ)・・・　２年生。部長。パソコンは詳しいけど，プログラミングははじめて。
「みぃちゃん」(みぃ)・・・２年生。唯一の女子。その大胆な発想にみんないつも驚き。
「よっさん」(よっ)・・・　３年生。頼りになる先輩。とてもやさしいアニキです。自称ゲーマー。
「ゆうちゃん」(ゆう)・・・１年生。パソコンにはあまり詳しくないけど，音楽ならまかせて。

## ～星音高校，パソコン研究部部室～

**よっ：**「はっ～くしゅん～！！」
**みぃ：**「あらら…花粉症？この時期になるといつもですね。はい，ティッシュ。」
**よっ：**「あ，ありがと…は，鼻が詰まって…。」
**ゆう：**「部長，ベーシックプログラミング教室読みましたか？」
**ちぃ：**「もちろん読んだよ！だんだん本格的になってきたね。新しい命令いろいろ覚えたから，早速ゲームに取り入れていくね。」
**みぃ：**「私も読んだよ～わくわくしてきたね。そうそう，ちょっと考えてみたの。見て見て。」

オープニング
↓
ゲームスタート　プレイヤー情報の入力
↓
メイン　謎解き／戦闘(経験値稼ぎ)／アイテム収集
↓
ボス戦　アイテム収集
↓
最終推理
↓
エンディング

**ちぃ：**「なるほど。ゲーム全体の流れだね。」
**みぃ：**「これだけじゃないよ。はい，基本のゲーム画面のデザイン。」
**みぃ：**「限りなくシンプルにまとめてみました。」



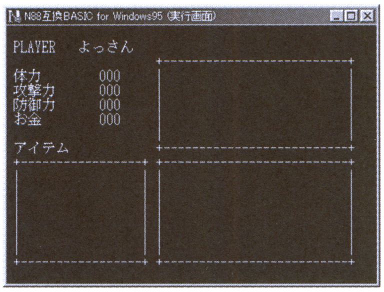

**ちぃ：**「うん。これなら今でもなんとかできそうかな。」
**みぃ：**「まだあるよー。はい，これがプレイヤー登録で，これがマップ…。」
**ちぃ：**「うわぁ～たくさんあるなぁ。」
**みぃ：**「うん。ストーリー考えながら作ったの。こういうのあったほうが，みんなも制作しやすいかなぁと思ったから。」
**よっ：**「そうそう仕様書は大事やで。ゲームの設計図みたいなもんやから,これがないと作業が進まない。みぃちゃんよく気がついたなぁ。」
**みぃ：**「まかせといて！！（なるほど仕様書っていうんだぁ…)」
**ちぃ：**「よし，じゃあ僕はこの仕様書をもとに早速プログラム始めるね。」
**みぃ：**「私はちょっと外へ…。」
**ゆう：**「みぃちゃんどこいくの？」
**みぃ：**「ゴミゴミしたこの部室じゃ，いいアイデアがでないから，海でも見ながらアイデアをまとめてきまーす！」
**ゆう：**「あらら…行っちゃった…。」
**ちぃ：**「(笑)」
**よっ：**「こっちものんびりしてる場合ちゃうで。どんどん作っていかないと間に合わへんよ…はっくしょーん！」
**ちぃ：**「そうですね。じゃ，基本のゲーム画面を作ってみます！」
**ゆう：**「よっさん先輩，大丈夫ですか…」

～20分後～

**ちぃ：**「できた！！今回はバグチェックばっちり

**(プログラムVersion0.02)**

```
10 '星音高校　パソコン研究部
20 '          プログラム制作開始　2002/3/1
30 '               Version 0.02      2002/4/1
100 CLS
110 PRINT "星音高校殺人事件（仮）"
120 PRINT ""
130 INPUT "主人公の名前を入力してください";PLAYER$
500 '基本画面
510 CLS
520 X1=1
530 Y1=1
540 LOCATE X1,Y1:PRINT "PLAYER   ";PLAYER$
550 LOCATE X1,Y1+2:PRINT "体力       000"
560 LOCATE X1,Y1+3:PRINT "攻撃力     000"
570 LOCATE X1,Y1+4:PRINT "防御力     000"
580 LOCATE X1,Y1+5:PRINT "お金       000"
590 'アイテムウインドウ
600 LOCATE X1,Y1+7:PRINT "アイテム"
610 LOCATE X1,Y1+8:PRINT "+-----------------+"
620 LOCATE X1,Y1+9:PRINT "¦                 ¦"
630 LOCATE X1,Y1+10:PRINT "¦                 ¦"
640 LOCATE X1,Y1+11:PRINT "¦                 ¦"
650 LOCATE X1,Y1+12:PRINT "¦                 ¦"
660 LOCATE X1,Y1+13:PRINT "¦                 ¦"
670 LOCATE X1,Y1+14:PRINT "¦                 ¦"
680 LOCATE X1,Y1+15:PRINT "+-----------------+"
690 'メインウインドウ
700 LOCATE X1+20,Y1+1:PRINT "+-------------------------+"
710 LOCATE X1+20,Y1+2:PRINT "¦                         ¦"
720 LOCATE X1+20,Y1+3:PRINT "¦                         ¦"
730 LOCATE X1+20,Y1+4:PRINT "¦                         ¦"
740 LOCATE X1+20,Y1+5:PRINT "¦                         ¦"
750 LOCATE X1+20,Y1+6:PRINT "¦                         ¦"
760 LOCATE X1+20,Y1+7:PRINT "+-------------------------+"
770 'メッセージウインドウ
780 LOCATE X1+20,Y1+8:PRINT "+-------------------------+"
790 LOCATE X1+20,Y1+9:PRINT "¦                         ¦"
800 LOCATE X1+20,Y1+10:PRINT "¦                         ¦"
810 LOCATE X1+20,Y1+11:PRINT "¦                         ¦"
820 LOCATE X1+20,Y1+12:PRINT "¦                         ¦"
830 LOCATE X1+20,Y1+13:PRINT "¦                         ¦"
840 LOCATE X1+20,Y1+14:PRINT "¦                         ¦"
850 LOCATE X1+20,Y1+15:PRINT "+-------------------------+"
```




です。エラーはでないです。」
**よっ：**「早速実行してみよう！ん？また名前入力するんやね。“よっさん”[ENTER]と…。お！画面でたねー。」
**ゆう：**「なんとなく雰囲気でてきましたね。」
**よっ：**「うん。だけど…」
**ちぃ：**「え，なんですか？今回ちゃんとバグチェックはしましたけど…。」
**よっ：**「そうじゃなくて。プログラムがやたら長いなぁ。今回の講座で習った**FOR～ NEXT**使ったらもっとシンプルになるんちゃうかなぁ。ウインドウの部分とか。」
**ちぃ：**「そういわれてみれば…。ん？そっか！確かに何か所か**FOR～NEXT**使えそうなところありますね。ほかにもいろいろと気がついたのでやってみます！！」

～10分後～

**ちぃ：**「できた！500行目以降を修正してみました。FOR～NEXTだけでなくIF～THEN～ELSEも使ってます。」

**(プログラムVersion0.03)**

```
10 '星音高校　パソコン研究部
20 '          プログラム制作開始　2002/3/1
30 '               Version 0.03      2002/4/1
100 CLS
110 PRINT "星音高校殺人事件（仮）"
120 PRINT ""
130 INPUT "主人公の名前を入力してください";PLAYER$
500 '基本画面
510 CLS
520 X1=1   '基本画面の左上X座標
530 Y1=1   '基本画面の左上Y座標
540 LOCATE X1,Y1:PRINT "PLAYER   ";PLAYER$
550 LOCATE X1,Y1+2:PRINT "体力       000"
560 LOCATE X1,Y1+3:PRINT "攻撃力     000"
570 LOCATE X1,Y1+4:PRINT "防御力     000"
580 LOCATE X1,Y1+5:PRINT "お金       000"
590 'アイテムウインドウ
600 LOCATE X1,Y1+7:PRINT "アイテム"
610 W1$="+-----------------+"   '上下ライン
620 W2$="¦                 ¦"   '左右ライン
630 FOR I=8 TO 15
640    LOCATE X1,Y1+I
650    IF I=8 OR I=15 THEN PRINT W1$ ELSE PRINT W2$
660 NEXT I
670 'メインウインドウ&メッセージウインドウ
680 W3$="+-------------------------+"   '上下ライン
690 W4$="¦                         ¦"   '左右ライン
700 X2 = 20                             '左X座標の設定
710 FOR I=1 TO 15
720 LOCATE X1+X2,Y1+I
730 IF I=1 OR I=7 OR I=8 OR I=15 THEN PRINT W3$ ELSE PRINT W4$
740 NEXT I
```

**よっ：**「随分すっきりしたね。ぱっと見た感じは，さっきのほうがわかりやすいけど，ウインドウサイズを後から編集する時なんていうのは，こっちのほうが楽やね。」
**ゆう：**「ウインドウの線はもっと綺麗にならないの？」
**ちぃ：**「うん。気にはなるんだけど，まだ習っ

てないから現時点ではこれで我慢して。」

**ゆう：**「そっかぁ。」

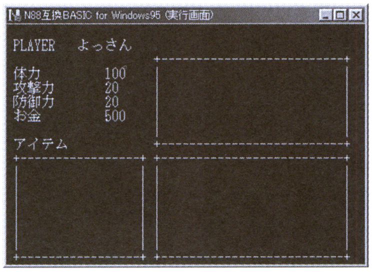
**ちぃ：**「よし，この勢いでプレイヤー登録もつくるぞ！」

～10分後～

**ちぃ：**「はい。これは，入力求めるだけだから簡単だった！」

**(プログラムVersion0.04) 580行目までのリスト**

```
10 '星音高校　パソコン研究部
15 '            プログラム制作開始　2002/3/1
20 '                 Version 0.04      2002/4/1
25 '初期設定
30 PLAYER$ = ""    '名前
35 HP=100          '体力
40 ST=20           '攻撃力
45 DF=20           '防御力
50 GD=500          'お金
55 TOSHI=0         '年齢
60 SEI=0           '性別
100 CLS
110 PRINT "星音高校殺人事件（仮）"
120 PRINT
130 INPUT "主人公のニックネームを入力してください";PLAYER$
140 INPUT "年齢を入力してください（半角数字）";TOSHI
150 INPUT "1=女，2=男";SEI
500 '基本画面
510 CLS
520 X1=1   '基本画面の左上X座標
530 Y1=1   '基本画面の左上Y座標
540 LOCATE X1,Y1:PRINT "PLAYER   ";PLAYER$
550 LOCATE X1,Y1+2:PRINT "体力        ";HP
560 LOCATE X1,Y1+3:PRINT "攻撃力      ";ST
570 LOCATE X1,Y1+4:PRINT "防御力      ";DF
580 LOCATE X1,Y1+5:PRINT "お金        ";GD
.
.
.
```




**よっ：**「どれどれ，ニックネームを入れて，年齢…性別…も入力…ＯＫ！」

**ゆう：**「おー！体力とか攻撃力とかのパラメータが表示されてる！！すごいー！」

**ちぃ：**「えっへん！」

**よっ：**「でも…年齢とか性別って入力したけど，どこにも使われてないよね。せっかくやし，年齢とか性別でパラメータを変化させてみたら?」

**ちぃ：**「というと…」

**よっ：**「男なら体力が2倍とかね。」

**ちぃ：**「なるほど！！それできそう！やってみます

**IF SEI=2 THEN HP=HP＊2**

これをどこかに入れれば…。」

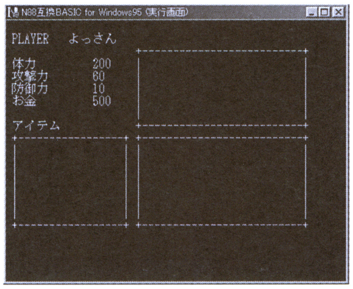
**よっ：**「せっかくやから，他のパラメータも調整してみたら。そうやな。攻撃力は3倍にして，防御力を1/2にしてみようか。」

**ちぃ：**「はい。ということは…

**IF SEI=2 THEN HP=HP*2:ST=ST,3:DF=DF/2**

よし！これを，160行目に入れて見ます。……

……よし！できた！先輩やってみてください。」




**よっ：**「ＯＫ！名前を入れて，年齢，性別…と。お！パラメータ変わってるねー。」

**ちぃ：**「やったぁ。ちょっとずつそれらしくなってきたぞ！」

**よっ：**「年齢によってもパラメータが変化するようにしよう。これは来月までの宿題！」

**ちぃ：**「はい！わかりました。」

**よっ：**「まあ，まだメインのゲーム部分ができてないから，パラメータの値は後でもう一度調整しないとあかんけどね。」

**ゆう：**「どうしてですか？」

**ちぃ：**「ゲームバランスだよ。男を選ぶと簡単で女を選ぶと難しい…とか，体力の値が少なすぎてすぐにゲームオーバーになる…とか。それじゃゲームとしてイマイチなので，最後に調整する必要があるんだよ。」

**ゆう：**「そっかーそうだね。」

**よっ：**「はっくしょーん！ふぅー。じゃ今日は俺，先に帰るわ。おつかれさ～ん。」

**ちぃ：**「あ，おつかれさまでしたー！」

**ゆう：**「おつかれさまでしたー！」

**ちぃ：**「ふと，思ったんだけど…よっさん先輩ってめちゃくちゃプログラム詳しくない？」

**ゆう：**「そういえばそうですね。こっそり予習でもしているのかな…。」

**ちぃ：**「まあ，いいか。それよりも，みぃちゃんが置いて行った仕様書がまだたくさんあるから，がんばって作っていくよ。ゆうちゃんもそろそろ曲作りにとりかかれそうかな？」

**ゆう：**「え，曲ですか？そ，そうですねー海でも見ながら考えてみます～！」

**ちぃ：**「あ！お，おい！なんだよ。みんないなくなったじゃないか…今日は僕も帰ろう。明日からがんばるぞ！」

（おいおい…）

こうして４人のゲーム制作第2回は終了した。誰もいなくなった部室…まさに前途多難なゲーム制作！果たして来月は…いったいどうなってしまうのか！！お楽しみに！！

**(プログラムVersion0.05)**

```
10 '星音高校　パソコン研究部
15 '            プログラム制作開始　2002/3/1
20 '                 Version 0.05      2002/4/1
25 '初期設定
30 PLAYER$ = "" '名前
35 HP=100        '体力
40 ST=20         '攻撃力
45 DF=20         '防御力
50 GD=500        'お金
55 TOSHI=0       '年齢
60 SEI=0         '性別
100 CLS
110 PRINT "星音高校殺人事件（仮）"
120 PRINT
130 INPUT "主人公のニックネームを入力してください";PLAYER$
140 INPUT "年齢を入力してください（半角数字）";TOSHI
150 INPUT "1=女，2=男";SEI
160 IF SEI=2 THEN HP=HP＊2:ST=ST＊3:DF=DF/2  '男を選んだとき
500 '基本画面
510 CLS
520 X1=1   '基本画面の左上X座標
530 Y1=1   '基本画面の左上Y座標
540 LOCATE X1,Y1:PRINT "PLAYER   ";PLAYER$
550 LOCATE X1,Y1+2:PRINT "体力        ";HP
560 LOCATE X1,Y1+3:PRINT "攻撃力      ";ST
570 LOCATE X1,Y1+4:PRINT "防御力      ";DF
580 LOCATE X1,Y1+5:PRINT "お金        ";GD
590 'アイテムウインドウ
600 LOCATE X1,Y1+7:PRINT "アイテム"
610 W1$="+-----------------+"  '上下ライン
620 W2$="¦                 ¦"  '左右ライン
630 FOR I=8 TO 15
640   LOCATE X1,Y1+I
650   IF I=8 OR I=15 THEN PRINT W1$ ELSE PRINT W2$
660 NEXT I
670 'メインウインドウ&メッセージウインドウ
680 W3$="+--------------------------+"  '上下ライン
690 W4$="¦                          ¦"  '左右ライン
700 X2 = 20                     '左X座標の設定
710 FOR I=1 TO 15
720   LOCATE X1+X2,Y1+I
730   IF I=1 OR I=7 OR I=8 OR I=15 THEN PRINT W3$ ELSE PRINT
W4$
740 NEXT I
```

好評発売中

**大学入学資格検定免除指定試験**

パーソナルコンピュータ利用技術認定試験受験者のための

2002年版 **パソコン認定試験③級④級**

**合格対策試験問題集**

出題傾向と解答・解説

パソコン認定３級，４級最新の出題問題と解答・解説が載っており，短期間に効率よく，しかも最短距離で合格できる。また企業内教育などに採用されている。

●Ｂ５判／定価2,100円（税込）

●お求めは全国書店または当社販売部，当社 総支局まで。

電波新聞社　TEL03-3445-8201

協力：（株）富士通南九州システムエンジニアリング

# はじめよう！ Microsoft Officeで BASICプログラミング nol.1

●文/椛島たけし

## 第1回 WordやExcelでできる!!

今回から新シリーズとして，Microsoft Office XPを使ってプログラミングをはじめます。プログラミングがはじめての人でも，クリックができれば大丈夫です。

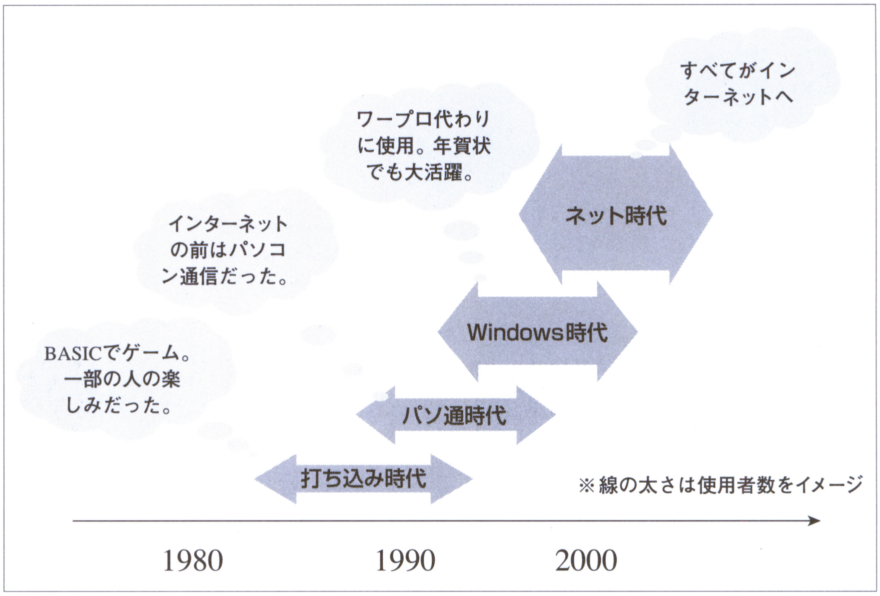
### プログラムはカンタンだ

春は合格・入学・進級・浪人……，祝いにパソコンを手に入れた人も多いのではないでしょうか。もちろん，入学した学校にパソコンが置いてあって，自由に？さわれるケースもあると思います。

「パソコン＝インターネット」というイメージをもっている読者も多いと思いますが(**第1図**)，携帯電話でもネットが楽しめる現在，インターネットだけにパソコンを使うのはもったいない！と思いませんか？。ホームページ作成なども魅力的ですが，いまいち当たり前ですね。

そこで，プログラミング(プログラムを作ること)にチャレンジしてみましょう。いろいろなソフトを使ってみることもそれなりに楽しいのですが，プログラミングには別世界の楽しさや達成感があります。また，モノを作る「創造力」を養うことにもつながります。

この連載では気楽に・安く・楽しくプログラムを作成するのがテーマです。難しく考えず，ぜひいっしょにやってみましょう。



◀《第1図》筆者が考える一般的なパソコンの使い方のイメージ

≪第1表≫主な言語とパソコンで動作する製品(一部)

| | | |
|---|---|---|
| **BASIC** | **昔からパソコンのメジャー言語。パソコン=BASICの時代もあった。** | |
| Visual Basic | マイクロソフト | もっとも使われているBASIC製品 |
| QBASIC | マイクロソフト | MS-DOS添付の英語版BASIC |
| F-BASIC | 富士通 | Windows対応のBASIC |
| N88-BASIC | 日本電気 | 主にPC-98(DOS)やROMで動くBASIC |
| **C/C++言語** | **UNIXが発祥の言語だが、パソコンで多く使われる。ちょっと難しいが複雑な処理も自由自在。** | |
| Visual C++ | マイクロソフト | C/C++言語のメジャー製品 |
| Visual C# | マイクロソフト | C++言語が進化した次世代のC言語 |
| C++Builder | ボーランド | C++のビジュアル開発ツール |
| **Pascal** | **もともと教育用に開発された言語。比較的やさしい。** | |
| Delphi | ボーランド | オブジェクト指向のPascal製品。 |
| Kylix | ボーランド | DelPhiのLinux版。 |
| **COBOL** | **企業の業務ソフトに昔から利用されてきた。パソコン、オフコン、汎用機のすべてで動作する。** | |
| PowerCOBOL97 | 富士通 | COBOLのビジュアル開発ツール |
| COBOL85 | (株)日立製作所 | COBOLのビジュアル開発ツール |
| Micro Focus COBOL | マイクロフォーカス | COBOLのビジュアル開発ツール |
| **Java** | **OSやハードウェアを問わず使えるように開発された言語。** | |
| JDK | サン・マイクロシステムズ | Javaを使う上で欠かせない製品 |

*Javaの開発製品については対応製品がかなり多いため、基本のJDKの紹介にとどめます

## BASICって

本誌は『マイコンBASICマガジン』です。つまり，BASICな雑誌なわけですね。BASICというのはプログラミング言語の一つです。ただし，製品を指すのではなく言語そのものを指します。プログラミング言語にはC言語，Java，COBOLなどさまざまなものがあります(**第1表**)。

とりあえず，BASICの特徴としては，「比較的簡単な英単語でプログラムが作成できるプログラム用の言葉」とカンタンに考えておきましょう。

## まずはパソコンの中身を見てみよう

Microsoft WordやMicrosoft Excelは，多くのパソコンに入っていると思います。Wordは文書を作成するときに使いますし，Excelはデータ集計やグラフ作成などに使われるソフトです(**第2図**)。一見プログラムとは関係なさそうですが，立派にプログラムを作ることができるのです。

## さっそくプログラミング

まずは，Microsoft Wordを起動し，[表示]-[ツールバー]とポイントしてください。「Visual Basic」というメニューがありますね。さっそくクリックしてみましょう。また，「コントロール ツール ボックス」というメニューもあります。こちらもクリックしてみましょう(**第3図**)。

次に，コントロールボックスのコマンドボタンをクリックし，Wordの文章の上にはりつけてください(**第4図**)。そしてコマンドボタンをクリックすると，「Microsoft Visual Basic」が表示されます(**第5図**)。

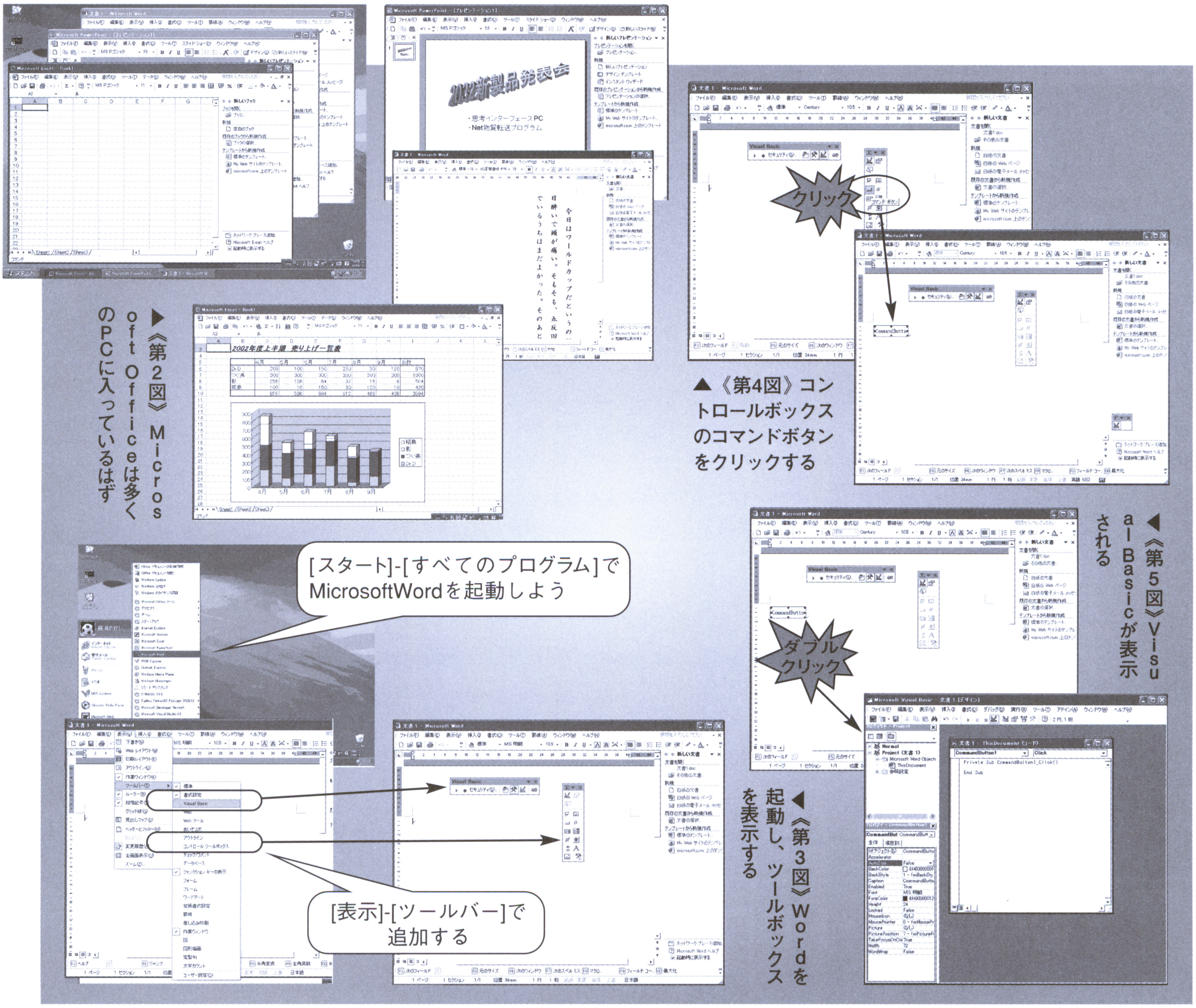
では，ここからが本格的？なプログラミングです。

```
Private Sub Command1Buttion1_Click()
```

の下に次の文字を入力してください。

```
MsgBox "マイコンBASIC Magazine"
```

入力が終わったら，タスクバーの文書をクリックして，文書を表示し，コマンドボタンをクリックしてみましょう




▶《第2図》Microsoft Officeは多くのPCに入っているはず

▲《第4図》コントロールボックスのコマンドボタンをクリックする

◀《第5図》Visual Basicが表示される

◀《第3図》Wordを起動し，ツールボックスを表示する

(**第6図**)。メッセージボックスが表示されますね。

ところで，MsgBoxとはメッセージボックスを表示する命令です。また，文字はダブルクォーテーション(")で括るのがBASICのルールです。

ここで，先ほどのプログラムをちょと変えてみましょう。

```
MsgBox Now
```

としてみましょう。今度は時間が表示されたはずです。

このように，プログラムを作ることは意外とカンタンなのです。

このプログラムはVBA(Microsoft VisualBasic Programming System Applications)を使ったプログラムです。Microsoft Office系の製品にはVBAと呼ばれるBASICをベースにしたツールがついており，プログラミングを行うことが可能なのです。

## 変数と代入について

ちょっとBASICの文法について説明します。

```
A = 1
```

とすると，数学ではAが1であることを示します。しかし，BASICではAという変数に1を代入するという意味になります。ですから，

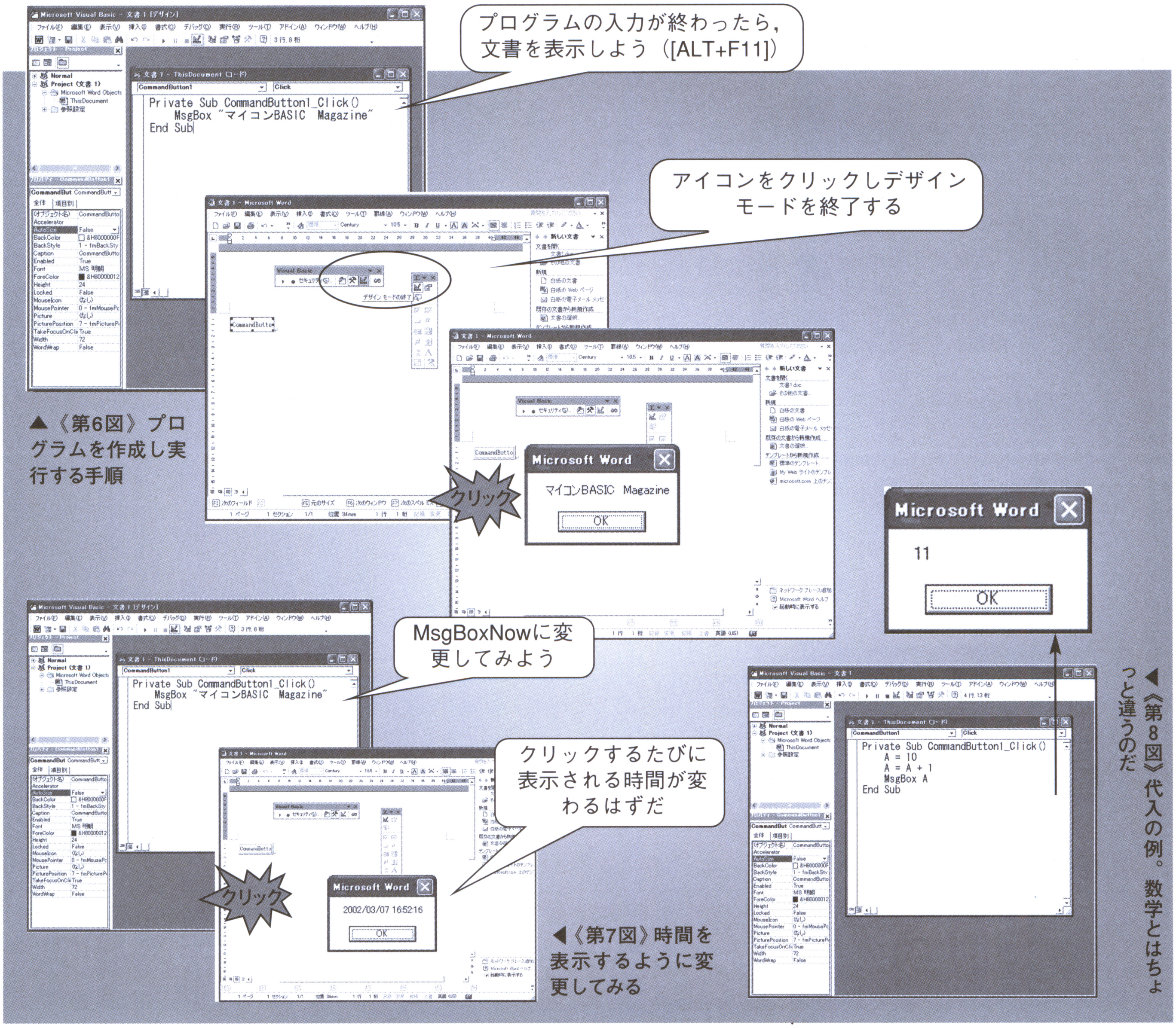
```
A = A + 1
```

という数学的にはおかしな書き方も，BASICでは正しいのです。この場合，「Aに対し，Aに1を加えたものを代入する」という意味になります。

```
A = 10
A = A + 1
MsgBox A
```

とすると，11が表示されるはずです(**第8図**)。

ここまでは，Wordでの例で説明しましたが，同じことが，Microsoft Officeのツールのすべてで実行することができます(**第9図**，**第10図**)。ここで，「それぞれの違いはあるの？」という疑問が出てくるでしょう。実は基本がVBA




▲《第6図》プログラムを作成し実行する手順

◀《第7図》時間を表示するように変更してみる

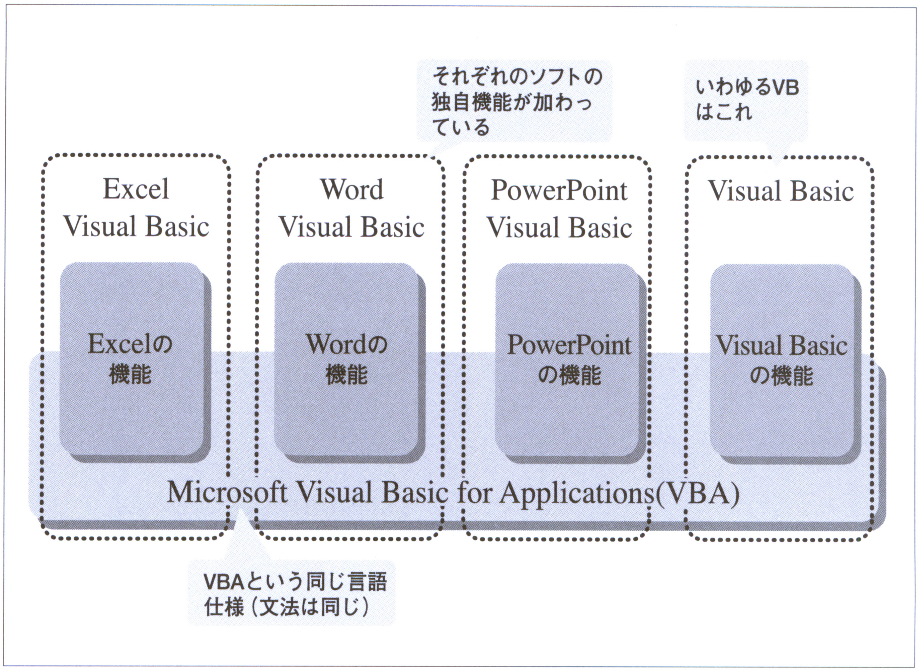
◀《第8図》代入の例。数学とはちょっと違うのだ

で，応用としてそれぞれのツールの機能があるのです(第11図)。

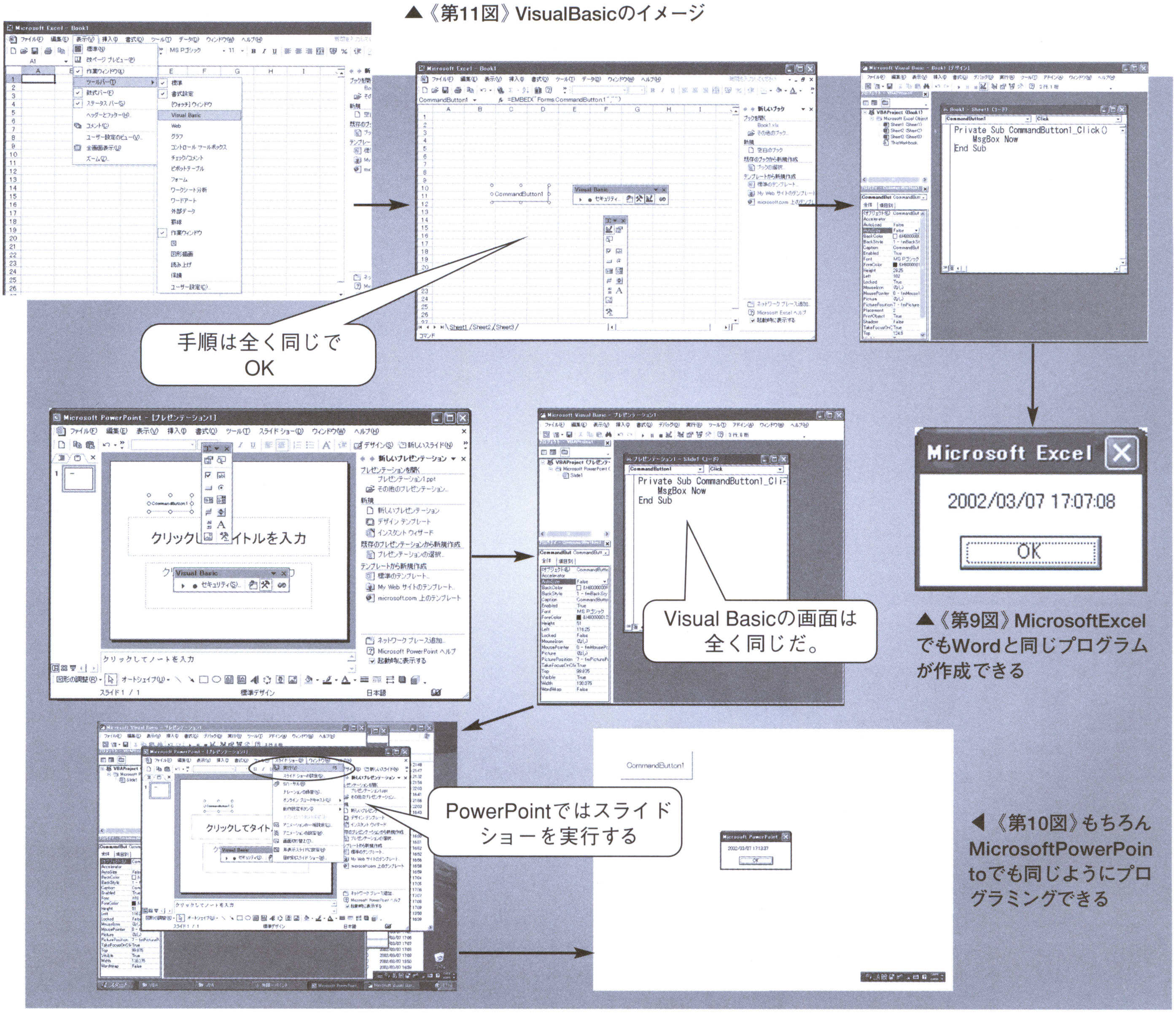
## どんなことができるのか？

さて，「カンタンなプログラムはつくれたけど，現実的にどんなプログラムが作れるの？」という疑問がわいてきませんか？とくに，どんなゲームができるかは興味があるところでしょう。

Microsoft Officeを使ったプログラムでは，激しいアクションは難しいですが，けっこう高度なゲームを作成することができます。プログラミング入門としては，手軽で最適なツールだと思いますよ。すこしずつ高機能なプログラムにチャレンジすることにしましょう。それでは来月もお楽しみに。




▲《第11図》VisualBasicのイメージ




▲《第9図》MicrosoftExcelでもWordと同じプログラムが作成できる

◀《第10図》もちろんMicrosoftPowerPointoでも同じようにプログラミングできる

■連載■ PCハードウエアテクニック

若松通商の「TX810EU」でパワフル・マシンに大変身!!

# 旧型AT機を完全復活 マザーボード換装テク

日増しにパワフルな進化をとげているDOS／V機ですが，以前自作したマシンを最新のCPUやHDDを載せ替えてアップグレードをはかろうとしても，対応できない場合も少なくないようです。とくにひと世代前のAT仕様のマシンでは，CPUのソケット形状が現行のものと異なっていたり，インターフェースが旧規格のままでパーツの性能が引き出せないことなどから，アップグレードが難しくなっています。今回は旧型自作マシンを，最新のCPUやHDD，メモリに対応したAT仕様のマザーボードを利用して，最新マシンに引けを取らない高性能PCへ変身させてみました。

## 1 「AT」と「ATX」規格の違いと制約

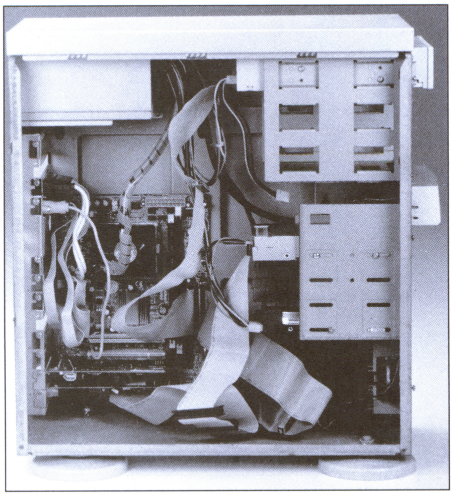
マザーボードの交換によるAT仕様パソコンのアップグレードを紹介する前に，少しDOS／V機の歴史をさかのぼってみることにしましょう。

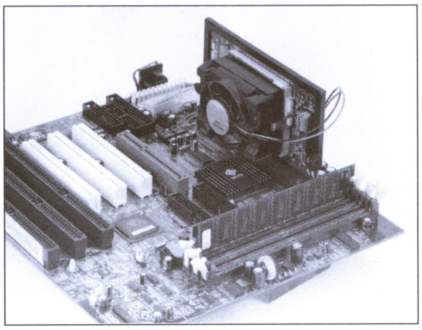
現在のDOS／Vパソコンは，世界最大手のコンピュータ・メーカー，IBMが最初に発売した個人ユーザー向けの小型コンピュータすなわちパソコンであるIBMパーソナルコンピュータ／ATが出発点になっています。

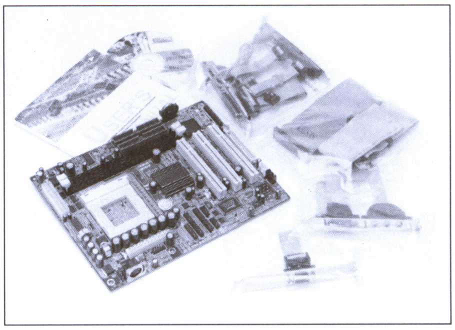
このAT機に採用された基本仕様をもとに，他のメーカーが発売しはじめデファクトスタンダードとして世界中に広まったのがPC／AT互換機で，それをDOS／V機とも呼んでいます。今回アップグレードを行うAT仕様のパソコンとは，PC／AT互換機に最初に採用された規格に準じた仕様を持つパソコンを指しています。

最近の自作DOS／Vパソコンは，ATX仕様が一般的です。ATXとは，その後のPCパーツの多角化や規格の高性能化に適合するようにATを進化させた規格（正確には規格ではなくフォームファクタと言う）で，最新のCPUやHDDなどのPCパーツの性能を最大に引き出せる設計が図られています。ATからATXへの移行で，それまで使えたパーツが規格が合わないため利用できなくなりました。PCケースやマザーボードは，ATとATXでは端子穴や取り付けネジの位置がまったく異なり，ATX仕様のパソコンへ流用することは無理。さらに電源電圧の違いからAT用の電源ユニットをATX仕様のパソコンに流用することも不可能です。

そこでなんとか使えるように，ということで登場したのが，AT仕様でありながら，ATXと同様に最新のインターフェース規格やCPUがそのまま使えるZIDA社の「TX810EU」マザーボードです。旧型の自作パソコンを末永く使いたい場合などには，とてもお得なマザーボードです。


▲AT仕様のパソコンでは，ATXに比べ使えるマザーボードやCPU，HDDなどのPCパーツの制約がある


▲AT仕様のマザーボードは，スロット1までのCPUに対応したものがほとんどで，最近のソケット型CPUは未対応


▲AT仕様でありながら，最新のCPUやメモリ，インターフェースが利用できるZIDA社の「TX810EU」マザーボード

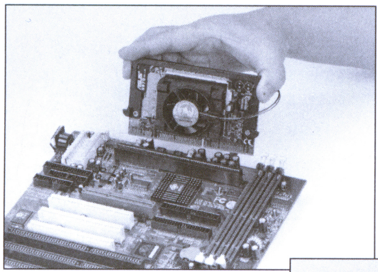
# 2 旧型AT自作マシンの解体とパーツの流用

マザーボードを交換する前に，まずATマシンから各PCパーツを取り外し，アップグレード機に流用できるものと，できないパーツを選別しましょう。耐用年数の長いマシンなので，ケース内部はホコリの巣になっているはずです。再び組み上げる前に掃除機などできれいにしておきましょう。

HDDやCD-ROMドライブそしてフロッピーディスクドライブに繋がっているケーブル類は，慎重に取り外してください。力まかせにケーブルを抜くと，ドライブの端子ピンが折れたり曲がることもあり，流用できるはずのパーツが最悪使えなくなる場合もありますので，要注意です。旧マシンからの流用パーツは，用意したオリジナル機のマザーボードが比較的新しいこともあり，メモリ，HDD，CD-ROMとフロッピーディスクの各ドライブの四つ。もし，Socket 370→Slot 1変換アダプタでPentium Ⅲまたは Celeron のCPUを搭載している場合は，オリジナル機のCPUがそのまま利用可能です。

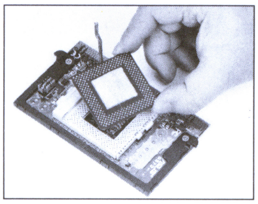
新規に高クロックCPUと交換する場合，FSBが100MHzまたは133MHzになるため，メモリもその周波数に対応したタイプを選ばなくてはなりません。なお，DIMM以前のSIMMタイプ・メモリは使えません。CPUとともに快適な動作性能の決め手となるのが，HDDです。AT仕様マザーの見られるE-IDEなどの旧インターフェース規格のHDDでも利用は可能ですが，転送速度が劣るためアップグレードにはUltra-DMA/33規格以上のHDDにしたいものです。

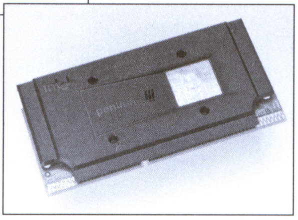
### ● AT機から流用可能なパーツ



●CPU：交換用のマザーボード「TX810EU」の対応CPUは，Socket370（PC-PGA,PC-PCA2）で，スロット1のCPUは流用できない。CPU変換アダプタでSocket370タイプのPentium IIIかCeleronなら使用可能。●メモリ：DIMMタイプのSDR SDRAMのみ。CPUのFSBに対応したメモリであれば流用可能。新しい製品であれば100MHzタイプでも133MHzに対応する場合もある。●HDD：数世代前のE-IDE規格でも利用可能だが，処理速度が低下するのでUltra-DMA規格に対応したHDDに交換しよう。●CD-ROMドライブ：読み出し速度が遅くなるが，そのまま利用可能。●フロッピー・ディスクドライブ：そのまま流用できる。

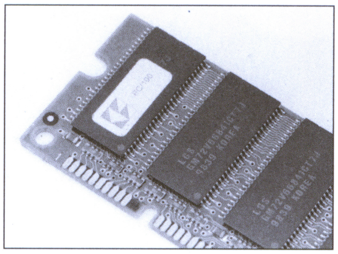
▲Socket370タイプのPentium ⅢまたはCeleronなら流用できる可能性は高い



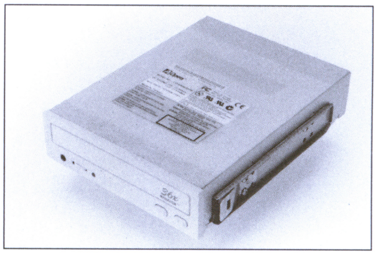


▲Slot 1タイプCPUは使えない

◀Socket370→Slot 1 CPU変換アダプタのCPUを取り外し流用

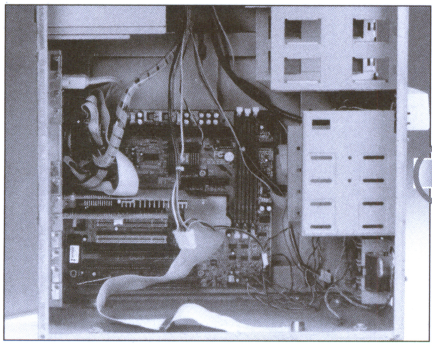


▲DIMM SDRAMでCPUのFSBに対応したメモリであれば流用可能

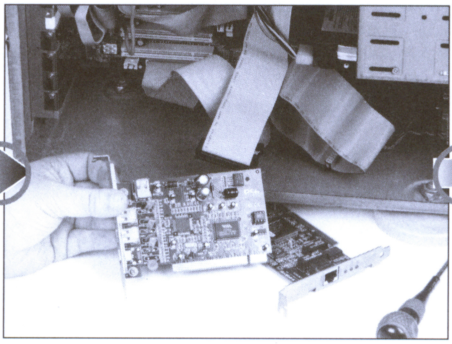


▲CD-ROM，FDドライブとも，オリジナルPCから流用可能

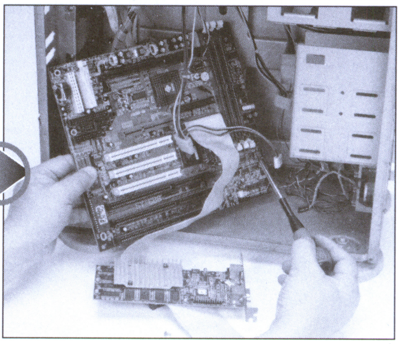
## ATマシンの分解とパーツ取り外し



▲HDD，CD-ROM，FDDにつながっている電源，接続ケーブルを取り外し，各ドライブを外す



▲オリジナルPCの拡張カードを慎重に外す。カードがPCIタイプなら，アップグレード機にそのまま流用可能



▲最後にオリジナルのマザーボードを取り外し，ケース内部のホコリなどをきれいに掃除する

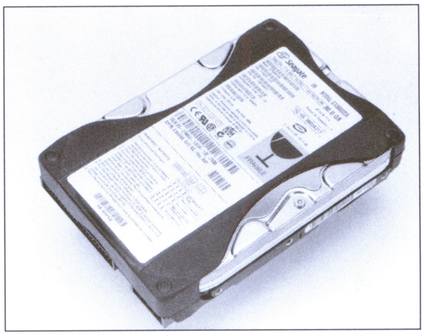
# 3 最新CPU・HDD・メモリ対応のマザーボード

今回のAT仕様パソコンでのアップグレードの核ともいえるのが，若松通商が販売している「Tomato」ブランドでお馴染みのZIDAテクノロジー社の「TX810EU」です。AT仕様のPCケースに完全対応のベビーATホームファクタのマザーボードで，ATマシンのアップグレードには，最適です。この種類の製品は，いまでは数少なく一般に入手できるのは，「TX810EU」以外には見あたらないようです。

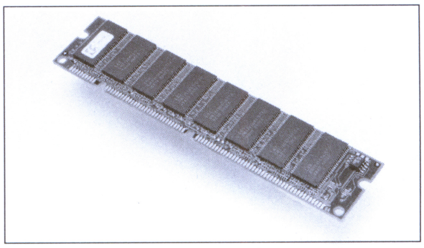
また，チップセットにインテル社の810Eを採用し，オンボードでAGPビデオグラフィックやサウンド入出力の各機能を備え，拡張ボードなしにUSBなどの最新インターフェースを持つDOS／V機へのアップグレードが簡単に行えるユニークな製品となっています。

CPUは，Socket370タイプでFC-PGAのPentium ⅢまたはCeleronに対応するほか，FC-PGA2のTualatinCPUの搭載も可能なことから，最新の高クロックCPUである1.3GHz Celeronや将来発売が期待される1.8GHz Celeronなどのギガバイト・クラスの高速CPUで，AT仕様にもかかわらず大幅なアップグレードが図れるスグレモノです。さらに，HDDとの接続規格は，Ultra-DMA（ATA）33／66が用意され，こちらも，最新の高速HDDが利用可能となっています。

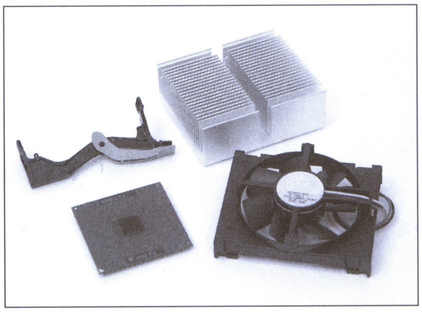
そこで，マザーボードの性能・機能を引き出すためのパーツを用意しました。CPUは駆動クロック1 GHzのCeleron，HDDにはシーゲイト社のAT340810A（40.8GB），256MB（PC100MHz対応）のほかに，ブロードバンドのインターネット接続用に，10Base-T／100Base-TX対応のLAN拡張カードも用意しました。このラインナップであれば，ハイエンドとまではいかないものの，かなり高速な動作速度が期待できるはずです。



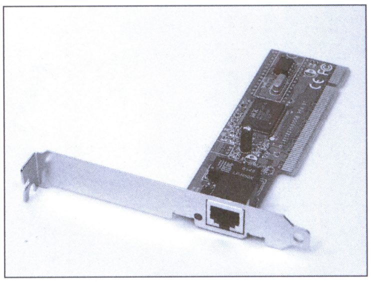
▲シーゲイト社ST340810は，Ultra-DMA／100規格に対応した最新HDDで，ラバー製の防震仕様が特徴



▲最新OSのWindows XPにも十分な容量の256MBタイプのSDRAMを選択。CPUのFSBに対応するPC100仕様

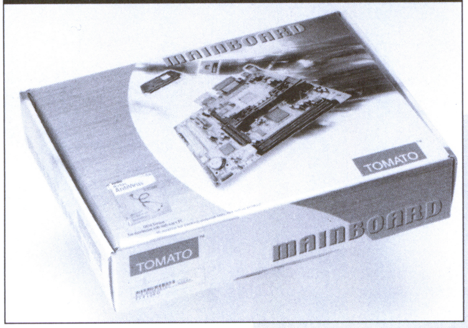


▲CPUにはギガヘルツクラスの1 GHz Celeronを用意。一般の用途であれば十分すぎるポテンシャルのCPUだ



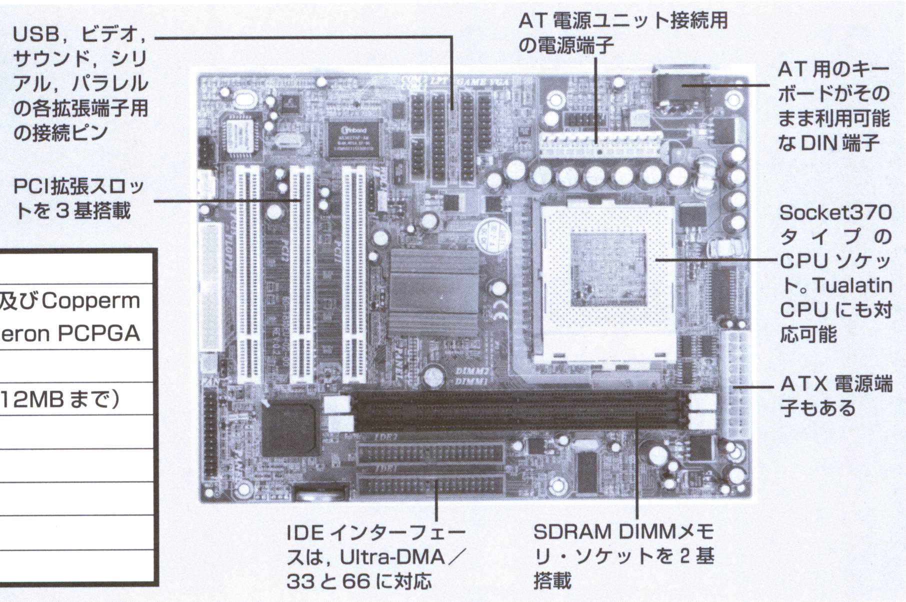
◀ブロードバンドによるインターネット接続も視野に入れてLANカードも用意

## ZIDA「TX810EU」




### 主な仕様

| | |
|---|---|
| ●チップセット | インテル810E |
| ●対応CPU | インテルTualatin FC-PGA2及びCoppermine FCPGA PentiumⅢ，Celeron PCPGA |
| ●FSB | 66／100／133MHz |
| ●メモリ | SDRAM DIMM×2（合計512MBまで） |
| ●拡張スロット | PCI×3 |
| ●IDE | Ultra-DMA33／66対応 |
| ●ビデオ | AGPオンボード |
| ●サウンド入出力 | AC97 2.1 オンボード |
| ●ホームファクタ | ベビーAT |



# 1 マザーボード換装＆マシンの組み立て

マザーボードによるアップグレードといっても，前ページまでのオリジナルAT機をパーツごとにいったん分解してしまえば，通常のDOS／Vマシン組み立てと，手順はほとんど一緒です。

注意すべきポイントとしては，今回交換するマザーボード「TX810EU」は，サウンド機能，ビデオグラフィック機能などが，オンボードで用意されていることから，各端子をPCケース背面のスロット・パネルに取り付けなければならないことです。

それでは，組み立て手順にそって解説していきましょう。

**●端子パネルとマザーボード接続配線**

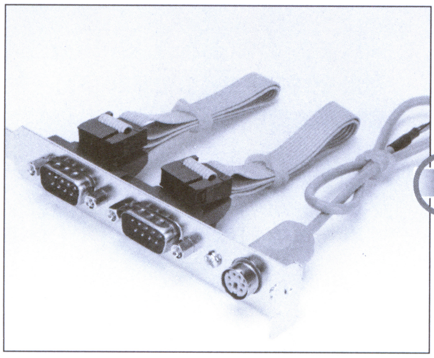
サウンド・ビデオ・グラフィックのほかにUSB，シリアル，パラレル，キーボード＆マウスなどの多くの端子をケーブルを介し，スロット・パネルに配置する「TX810EU」は，PCケースにマザーボードを取り付ける以前に，端子ケーブルをマザーボードへ予め配線しておいたほうが作業が進めやすいようです。

ただし，接続端子がマザーボード上のPCIスロットとCPUソケット間に集中して配置され，どれも同様なピン形状なので付属マニュアルで確認をとりながらケーブルを配線してください。

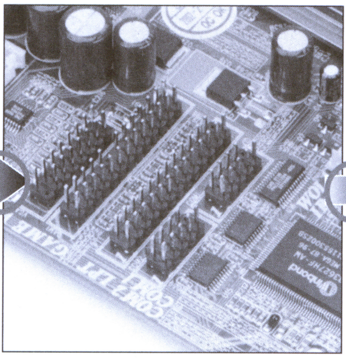
なお，「TX810EU」にかぎらず，数多くのマザーボードの付属マニュアルは英文表記となっており，英語が苦手な人には少々取っつきにくいかもしれません。しかし，文章に出てくる単語は，難解なものはわずかです。中学生レベルの解読力で心配ないはずです。わからない専門用語などがあった場合は，英和辞書で調べるか，学校の先生に教えてもらってください。

英文のマニュアルにチャレンジすることで，英語の解読力のアップとパソコンのアップグレードの一石二鳥が図れますよ!!

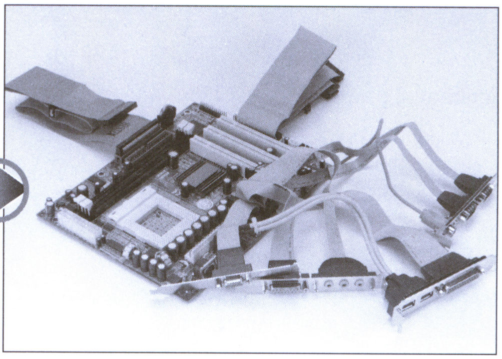
## i 端子パネルとマザーボード接続配線



▲ビデオ出力，サウンド入出力，USB，シリアル，パラレル，マウスなどの端子はスロット・パネルへ拡張するスタイル

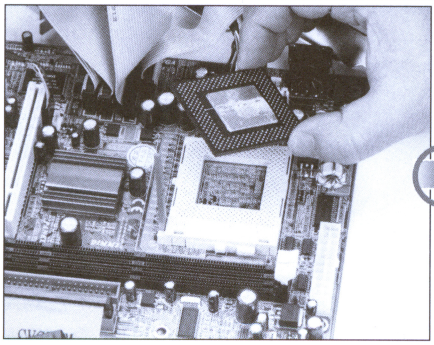


▲マザーボード上の拡張端子のケーブル接続用ピン端子。似た形状なので間違えないように!!

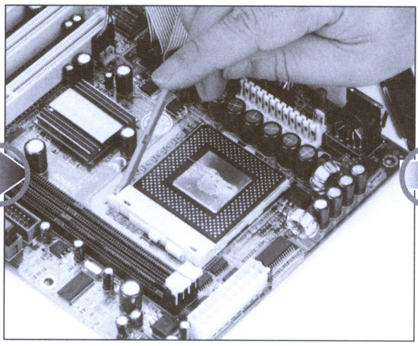


▲拡張端子パネルは，PCケースにマザーボードを組み込む前にケーブル配線を済ましておいたほうが，組み立てはラクだ

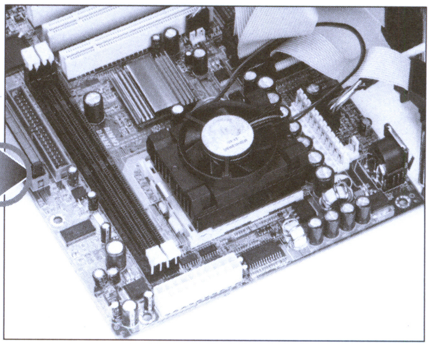
## CPU＆冷却ファンの取り付け



▲FC-PGAソケットへCeleron CPUをピン方向を確認し装着。ロックレバーは引き起こしておくこと



▲CPUをソケットへ装着したら，ロックレバーを水平方向に倒し，ピンをロックする



▲CPUに冷却フィンとクーリング・ファンを装着。CPUコア部分とフィンの接触面が所定位置にセットされているかを確認

●CPU&冷却フィン&ファンの取り付け

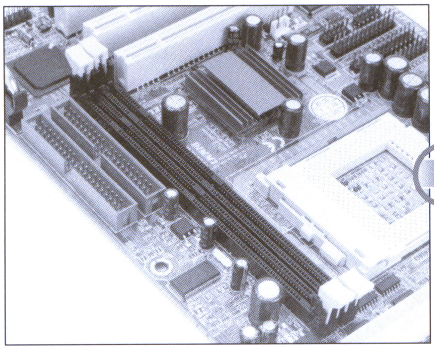
パソコンの自作に慣れた人なら問題なくできることも，慣れすぎて手を抜いたり，肝心なことを忘れてしまう場合が意外に多いものです。CPUまわりの作業は，いつも初心者のつもりで慎重に進めてください。

とくに，今回のFC-PGAソケット・タイプのCPUでは，冷却フィンとCPUのコア部分の位置がずれ，効率よく廃熱できない場合もあります。冷却能力が低下すると，動作が不安定になったり，エラーの発生やフリーズ（動作不能）の原因にもなります。CPUへのフィン取り付け後は，正常な位置にセットされているか，確認しておきましょう。なお，CPUとフィンの導熱性を高める特製シリコン・シートやグリスなどが市販されています。フィン側にシリコン・シート（白く薄いシール）が貼られている場合は，そのまま利用しても十分な放熱効果が得られます。市販のシリコン・グリスなどを用いる場合は，接触部分からはみ出さないようにしてください。心配なら使わない方が安心です。

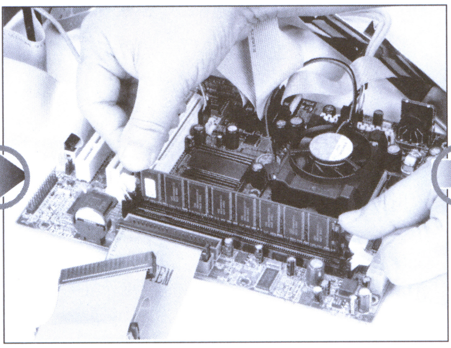
ところで，PCショップで販売されているCPUは，リテール版とバルク版の2種類があります。リテール版はインテル社などが正式の流通ルートで販売しているパッケージ付きの製品です。もう一方のバルク版は，メーカーのセットアップ済みパソコンへの組み込みを前提としたCPUで，パッケージ販売は行われていません。リテール版よりもバルク版のほうが安価で入手できるのが一般的です。ただし，リテール版は，CPUの仕様に適合した純正の冷却（廃熱）ファンが付属しているものの，CPU単体のバルク版では，別に冷却ファンを購入しなければなりません。単に安いという理由だけでバルク版に手を出すと，かえって高くついてしまうこともありますので，注意してください。

メモリの装着，PCケースのケーブル配線などは，慎重に行えばとくに問題はないはずです。なお，「TX810EU」マザーボードは，AT仕様のほかATX仕様の電源ユニットにも対応しています。ATX電源に交換すれば，AT電源ではできなかった自動電源オフ（ジャットダウン時）が行えるようになります。

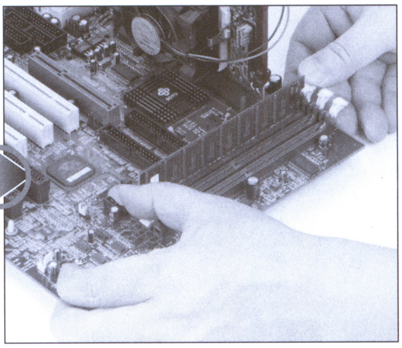
## SDRAM DIMMメモリの装着



▲マザーボード「TX810EU」は，SDRAM DIMMメモリを2枚まで装着可能だ

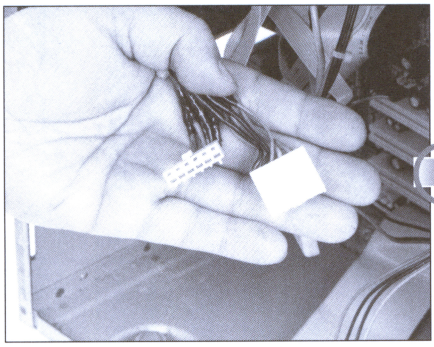


▲メモリは左右均等に力を加えて，スロットへ押し込む。所定の位置にセットすると，両サイドのロックレバーで自動固定する

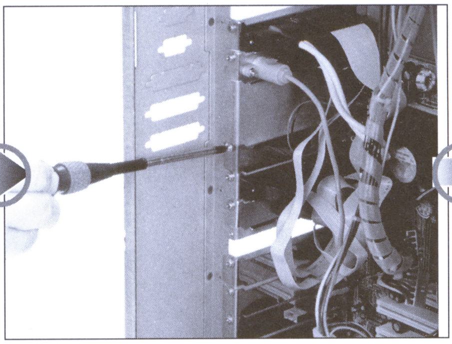


▲メモリの取り外しは，両サイドのロックレバーを押し下げるだけ。決してメモリを引っこ抜かないように!!

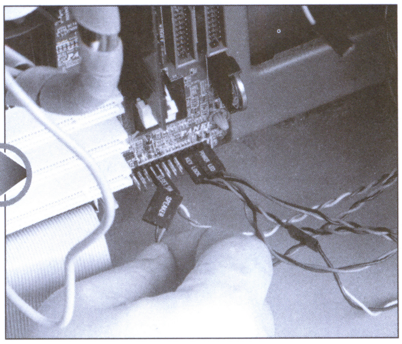
## 電源ケーブル&PCケース・ケーブルの接続



▲くし形のAT仕様電源ケーブルをマザーボードのAT電源端子へ接続。ATX電源にも対応する

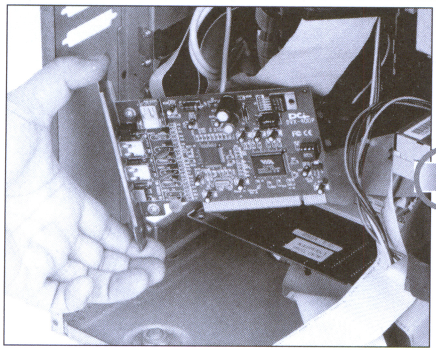


▲PCケース背面のスロット・カバーを取り外し，あらかじめマザーボードに接続しておいた拡張端子パネルを取り付ける

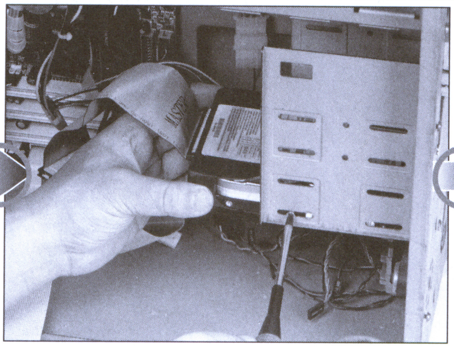


▲PCケースの配線ケーブルをマザーボードの接続ピンへつなぐ。細かい作業なので，誤配線には要注意

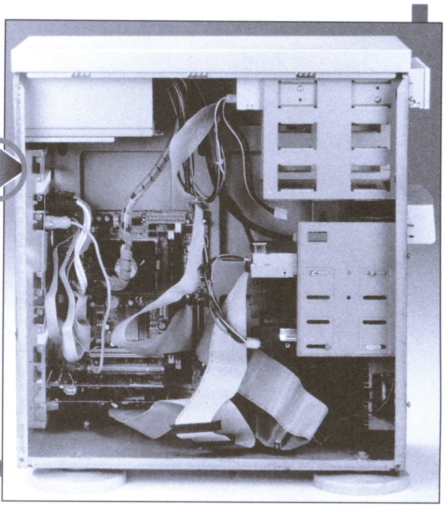
## 拡張ボード・HDD類のセットアップで完成



▲オリジナルATマシンで使用していた拡張カードをPCIスロットへ再びセット。3枚までの拡張が可能



▲CD-ROM，FDD，HDDの各ドライブへケーブル接続



▶コンパクトなベビーATのマザーのおかげでPC内部はすっきり

### 最新マシン並の体感速度!! 数年は活躍できる性能だ

さて，組み上がったATアップグレード・マシンですが，その性能はやはり1GHz駆動のCeleron CPUが大きく貢献していることもあり，オリジナルAT機とは比べものにならないほどの軽快さが得られました。メモリのデータ転送速度の向上に繋がるFSBも，従来の66MHzから100MHzへ一挙にアップ。最強の機能ながら高速な処理能力が要求される最新ＯＳ，Windows XPもサクサク操作できるようになりました。

手持ちにＡＴマシンがあり，そのまま長持ちさせたいという人には，今回の「TX810EU」マザーボードによるアップグレードは，ベストな選択でしょう。これで，あと数年は愛機が現用機として活躍してくれるはずです。

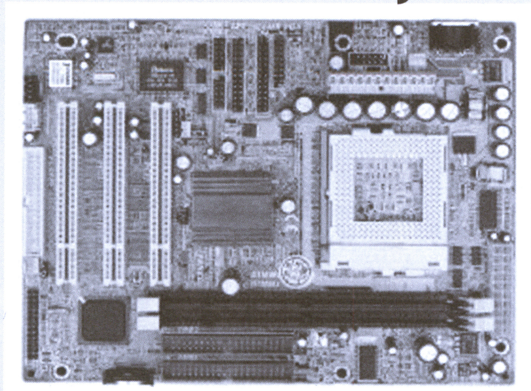

eマート http://www.wakamatsu-net.com eマートの価格は広告掲載の価格と一部異なります。

# 大好評ZIDAのATマザーシリーズ! ついにTualatin対応です。

大人気!

ZIDA TECHNOLOGY

**TX810EU ￥9,800**

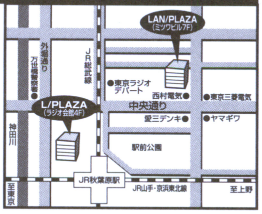


現行Celeron 1.3GHzに対応。今年の後半に出るという噂のCeleron 1.8GHzにも 対応か?(現時点では未確認)。486や初代Pen90の皮をかぶったGIGAマシーンを作りたいあなたに! 初期の自作派の方へお勧めします。
(AT互換機用ですので、NECの9821シリーズ等には対応しておりません。)

- ●Intel Tualatin FC-PGA2対応/ Coppermine FCPGA Pentium Ⅲ, Celeron FCPGA 対応
- ●FSB 66/100/133MHzサポート
- ●intel 810E chipset + ICH 1 Chipset
- ●DIMM slots x 2 装備(合計512MBまで、PC100使用)
- ●Auto Jumper
- ●PCIスロット x 3、USBポート x 2、IrDAポート x 1
- ●FDDポート x 1、LPTポート x 1、w/ ESD/EMI protection、COMポート x 2
- ●WOLコネクタ x 1
- ●ハードウェア温度・電圧モニタリング
- ●デュアル IDE チャンネルサポート、Ultra-DMA 33/66 IDE デバイス x 4
- ●Integrated AC97 オーディオオンボード
- ●AC97 2.1 Compliant Codec with 3D Stereo Enhancement
- ●1 x ラインアウト、1 x ラインイン、1 x マイクイン
- ●1 x CD-in、1 x AUX-in、1 x Game ポート
- ●キーボード・マウス過電流保護
- ●ATX 電源供給インターフェース
- ●Integrated オンボード VGA
- ●Baby AT (寸法 220mm x 170mm)

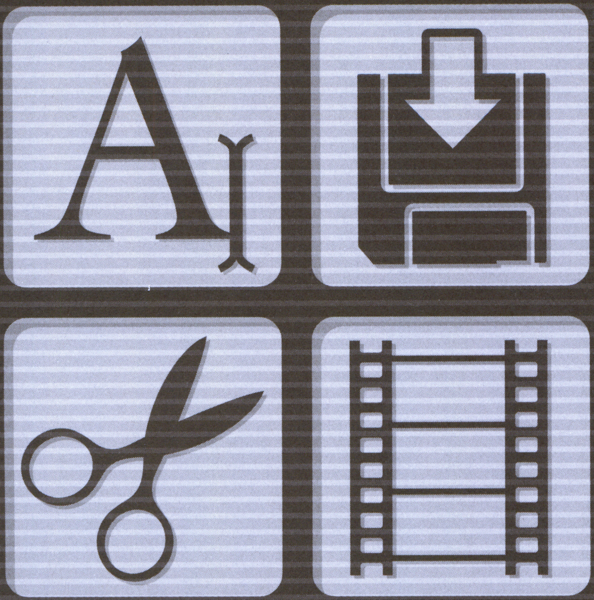
PC Customize PROSHOP 若松グループ http://www.wakamatsu.co.jp/ TEL03-3257-0601 FAX03-3251-7877
LAN/PLAZA (ミツワビル7F)
〒101-0021 東京都千代田区外神田1-11-4ミツワビル7F
■営業時間AM10:00〜PM7:00
■銀行振込口座:三和銀行 秋葉原支店 当座311336 口座名義(株)若松通商

PC Customize PROSHOP 若松グループ http://www.wakamatsu.co.jp/lplaza/ TEL03-3251-5054 FAX03-3251-5467
L/PLAZA (ラジオ会館4F)
〒101-0021 東京都千代田区外神田1-15-16ラジオ会館4F
■営業時間AM10:00〜PM7:00
■銀行振込口座:東京三菱銀行 秋葉原支店 当座9018950 口座名義(株)若松通商




株式会社 若松通商 2002. 3/29

※広告の価格には若干の変動がございます。ご了承ください。
通信販売の場合は送料着払となります。
※すべての商品に消費税が加算されます。



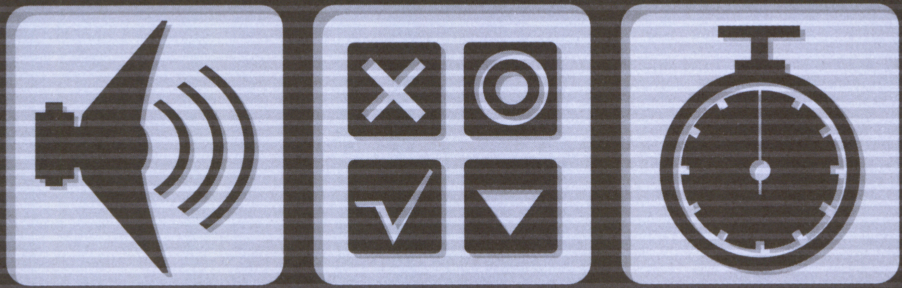
本コーナーは，さまざまな言語に対応したプログラミング学習と創作をテーマに，読者からの投稿作品で成り立っています。これから，プログラミングを勉強しようというみなさんや，プロのプログラマー／クリエイターを目指そうというみなさんにとって役立つ情報が満載です。手近にあるパソコンに入力して，お楽しみください。

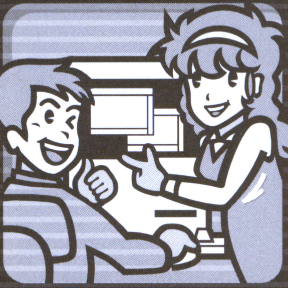
それぞれの記事にはプログラムを入力／実行するときの注意点やプログラムの内容，改造方法などが紹介されていますので，よく読んでください。また，“CHECKER FLAG”というコラムがあり，デバッグ博士ことDr.Dがそれぞれのプログラムに対し批評を加えています。口が悪い人なので，かなりキビシイことも言っていますが，自分でプログラムを作ったり，これから投稿しようとしている人にとっては，とても参考になるでしょう。

- Visual Basic
- Visual Basic +DirectX
- VBScript +DirectX
- Excel（VBA）
- F-BASIC
- HSP
- Javaアプレット
- iアプリ
- PIECE

# BM特選 投稿プログラムコーナー

## プログラマー＆クリエイターへの登竜門





## プログラム投稿者へのお願い

自分の作ったプログラムを投稿しようという人はつぎのきまりを守って応募してください。

**①プログラム（作品）ファイル ②プログラムの説明，住所・氏名・年齢・職業（学校名）・電話番号（連絡のつく時間も）を書いた原稿のテキスト・ファイルをフロッピーディスクなどにセーブし（プリンタで打ち出した原稿も同封のこと），お送りください。**

プログラムの説明には，①内容 ②使いかた・遊びかた ③操作方法 ④プログラムのしくみ（変数表やフローチャートも）⑤プログラムの具体的入力方法，注意点 ⑥改造のアドバイス，作ったとき苦労した点 ⑦使用機種名 ⑧言語（バージョン，モードなども）などを書きます。

**注1**）プログラムは必ずチェックし，ミスを防ぐために同じものを2～3入れてもらえば安心です。

**注2**）プログラムの説明は，テキスト・ファイルをフロッピーディスクなどにセーブし，念のためプリンタで打ち出したものも同封することが基本ですが，手書き原稿でも受け付けます。その際，できるだけていねいに，マス目のある用紙を使用してください。

**注3**）ディスクのラベル，封筒の裏面にもプログラム名（ファイル名も），原稿ファイル名，住所，氏名，電話番号を必ず記入してください。

内容はオリジナルのもので，未発表のものに限ります。もし，参考にした記事や本がありましたら，書名，号数，記事名，筆者名，記事のタイトルなども明記しておいてください。

お送りいただいたディスクはお返しできませんので，あらかじめバックアップをとっておいてください。

掲載されたプログラムに対しては，規定の原稿料をお支払いし，掲載誌をお送りいたします。

**編集部**

★本誌とじ込みの愛読者カードについている応募カードも忘れずに同封してください。

**あて先：〒141-8715 東京都品川区東五反田1-11-15 電波新聞社マイコンBASICマガジン「特選プログラム・コーナー」係**

**掲載プログラムのダウンロード／投稿はこちら**

**http://www.basicmagazine.net/**

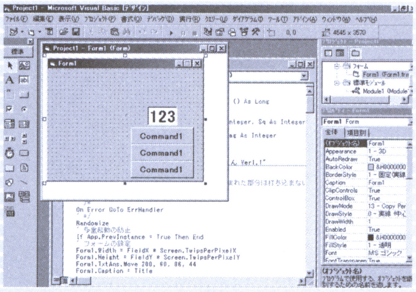
掲載[投稿プログラム]言語

# 入手方法と入力手順ガイド

## Visual Basic

### 製品紹介

Visual Basic(以下VB)とは，マイクロソフトから発売されている，Windows上で動作するプログラム作成ツールです(開発画面は**写真1**)。現在はVersion6.0が最新版となっています。また，EXE（実行ファイル）作成などの機能を制限した評価版(体験版)も存在します。



▲写真1

### 入手方法

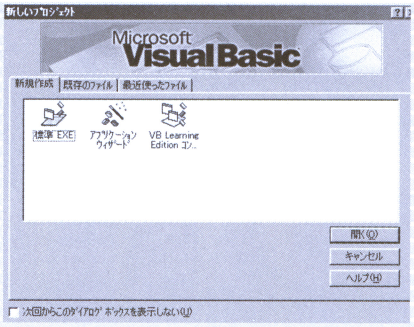
製品版は，各種類ともパソコンショップなどで購入可能です。学生・学校職員向けに価格を抑えたアカデミックパッケージも用意されています。

また，評価版は本誌で連載していた記事をまとめたムック本『入門Visual Basic　6.0/5.0講座』(税込み2100円)をはじめ，いくつかの書籍に付録として収録されています。

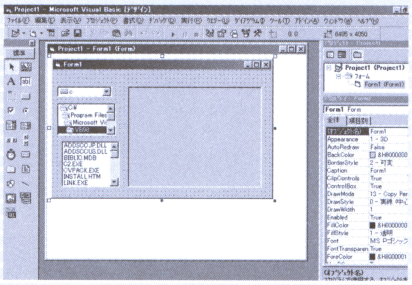
### 本誌プログラムの基本入力方法

①VBを起動後，標準EXEを選択します(写真2)。

②掲載プログラムのフォームの配置図を



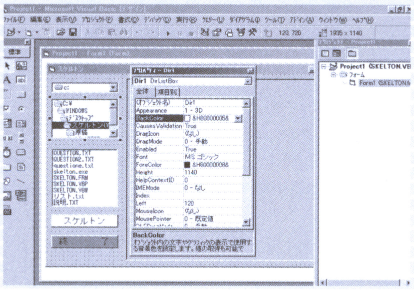

▲写真2



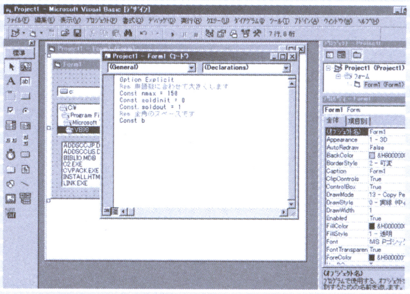
▲写真3

参考にしながら，オブジェクトをフォーム上に配置します(写真3)。

③掲載しているプロパティ表にそって，配置したオブジェクトのプロパティを設定（すでに入っている値を変更）していきます(写真4)。プロパティによっては，別に作成した画像ファイルなどを指定す



▲写真4



▲写真5

る場合がありますので，その場合は本誌に掲載されている画像写真を参考に，Windows付属の「ペイント」などのソフトを用いて画像を作成します。

④フォームのコードウィンドウを表示し，掲載されているプログラム・リストを入力していきます(写真5)。モジュールのリストがある場合は，同様にモジュールのコードウィンドウからリストを入力していきます。

⑤フォームが複数ある場合は，そのフォームの数だけ，②～④の作業を繰り返します。

⑥すべての入力作業が終了したら，実行ボタンを押しプログラムを実行させます(写真6)。

▲写真6

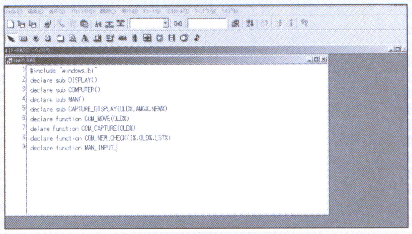
# F-BASIC(Windows版)

### 製品紹介

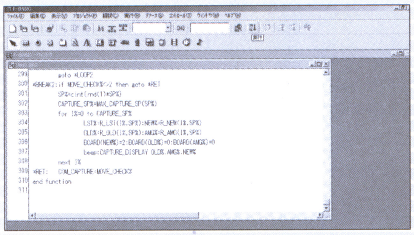
F-BASICとは，富士通ミドルウェアから発売されているWindows用プログラム言語です。現在はVersion6.3が最新版となっています。

### 入手方法

パソコンショップなどで購入可能です。



▲写真1



▲写真2

### 本誌プログラムの基本入力方法

①F-BASICを起動後，画面上のエディタから，掲載されているリストを入力します(写真1)。

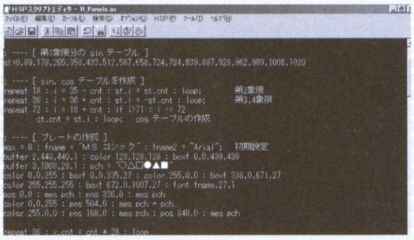
プログラムによっては，別途画像ファイルを必要とする場合があります。その場合は，誌面に掲載されている画像写真を参考に，Windows付属の「ペイント」などのソフトを用いて画像ファイルを作成します。

②リストを入力後，実行ボタンを押してプログラムを実行させます(写真2)。

# HSP

### 製品紹介

HSPとは，フリーで使用できるWindows用プログラム言語です(開発画面は写真1)。現在はVersion2.55が最新版となっています。



▲写真1

### 入手方法

HSPは，以下のホームページからダウンロードすることができます(写真2)。

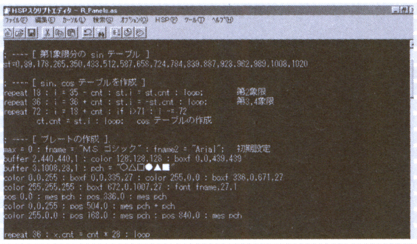
http://www.onionsoft.net/hsp/

また，パソコン雑誌やHSP関連の書籍などの付録として収録されていることもあります。

### 本誌プログラムの基本入力方法

①HSPを起動後，画面上のエディタから，掲載されているリストを入力します。

プログラムによっては，別途画像ファイルを必要とする場合があります。その



▲写真2

場合は，誌面に掲載されている画像写真を参考に，Windows付属の「ペイント」などのソフトを用いて画像ファイルを作成します。

②リストを入力後，実行ボタンを押してプログラムを実行させます。

# iアプリ

### 製品紹介

iモード対応の携帯電話を使用し実行します。プログラミングはパソコンで行います。

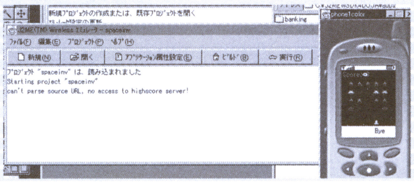
### 入手方法

携帯電話ショップで購入可能です。

### 本誌プログラムの基本入力方法

①サンマイクロシステムズのWEBサイト（http://java.sun.com/）からJava2の開発環境「Java 2 SDK Standard Edition 1.3.0」をダウンロードします。インストールすると“C:¥jdk1.3”というディレクトリにJavaコンパイラが展開されます。

② NTTドコモのWEBサイト（http://www.nttdocomo.co.jp/i/java.html）からiアプリの開発ツール「J2ME Wireless SDK for the DoJa release2.2（以下，DoJa)」をダウンロードします。ダウンロードには名前の登録が必要です。インストールすると“C:¥J2MEWSDK4DOJA”フォルダに主要なファイルが展開されます。



▲写真1

③スタートメニューから「J2ME~DoJa」を選択して，「KToolBar」を起動します(写真1)。

④「新規」ボタンを押して，新規プロジェクト名を入力すると，“C:¥J2MEWSDK4DOJA¥apps”フォルダの下にプロジェクト名のフォルダと,その下に“bin”,“res”,“src”フォルダが作成されます。

DoJaは，テキストエディタを搭載していないので，別途メモ帳などのテキストエディタソフトでソースファイルを作成します。入力方法は，誌面に掲載されているリストを入力し，指定されたファイル名で保存します。ファイル名を間違えると正しくビルドされないことがあります。また，DoJaの場合，jamファイルは

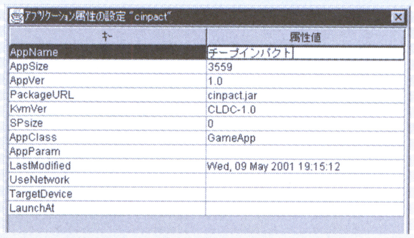
次のページにつづく

ソフト側で作成できるため，入力する必要はありません。

プログラムによっては，別途画像ファイルを必要とする場合があります。その場合は，誌面に掲載されている画像写真を参考に，Windows付属の「ペイント」などのソフトを用いて画像ファイルを作成します。

⑤“res”フォルダに画像（gif）や音楽ファイル（mld），“src”フォルダにはソースファイル（java）を格納します。

⑥「アプリケーション属性設定」ボタンを押して，キーの値を入力します(写真2)。（“AppSize”と“LastModified”は自動的に更新されます)。


アプリケーション属性の設定 "cinpact"

| キー | 属性値 |
|---|---|
| AppName | チープインパクト |
| AppSize | 3559 |
| AppVer | 1.0 |
| PackageURL | cinpact.jar |
| KvmVer | CLDC-1.0 |
| SPsize | 0 |
| AppClass | GameApp |
| AppParam | |
| LastModified | Wed, 09 May 2001 19:15:12 |
| UseNetwork | |
| TargetDevice | |
| LaunchAt | |

▲写真2

AppName = (アプリケーション名)
AppVer = 1.0
AppClass = (最初に実行するクラス名)
SPsize = (スクラッチパッドのサイズ)
KvmVer = CLDC-1.0

⑦「ビルド」ボタンを押すと，コンパイルとプリベリファイが一括して行われ，jarファイルとjamファイルが“bin”フォルダに生成されます。jarファイルは“res”フォルダの中身とクラスファイルがzip形式で圧縮されたものです。

⑧エディタで“C:¥J2MEWSDK4DOJA¥lib¥devices¥phone1color”フォルダの“phone1color.properties”を以下のように書き換えます（値は120×130ピクセル時の設定です)。書き換えたら「編集ー設定」をメニュー選択して，「デバイス指定」を“phone1color”にします。

```
    :
screen.x=27
screen.y=65
    :
screen.width=120
screen.height=156
    :
icon.up: 44, 138, off
    :
icon.down: 44, 146, off
    :
screenPaintableRegion.width=120
screenPaintableRegion.height=130
    :
softbutton.0=0,138,36,18, softButton
softbutton.1=90,138,36,18, softButton
    :
```

▲写真3

⑨「実行」ボタンを押すと，携帯電話エミュレータ上でｉアプリが実行されます(写真3)。

# VisualBasic + DirectX

## 製品紹介

VisualBasic + DirectXとは，本誌ではVisualBasicのプログラムで，DirectXを使用しているプログラムを示します。実行には，VisualBasicの他にDirectX SDKというソフトウェアが必要になります。

## 入手方法

VisualBasicの入手については，VisualBasicの項目を参照してください。

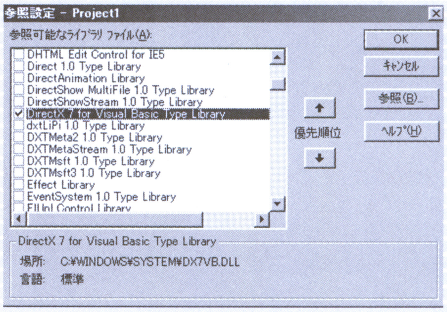
DirectX SDKは，マイクロソフトのHPからの無償ダウンロードや，パソコン雑誌のCD-ROMに収録されているものから入手することができます。

DirectX SDKの，現在のVersionは8.0aで，VisualBasicで使用するには，7.0以上のVersionである必要があります。

## 本誌プログラムの基本入力方法

① 基本的には，VisualBasicの項目で説明している方法と，同じ方法で入力していきます。

ただし，入力前に右記の設定を行う必要があります。



▲写真1

・［プロジェクト］メニューの［参照設定］で「DirectX7 for Visual Basic Type Library」にチェックを付ける(写真1)。

# VBScript+DirectX

## 製品紹介

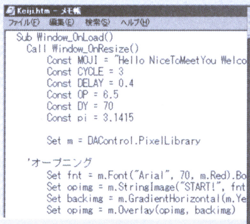
VBScript+DirectXとは，Internet Exporlerで動作するスクリプト言語(VBScript)でDirectAnimationというDirectX技術の一つを利用して動作するプログラムを示しています。

## 入手方法

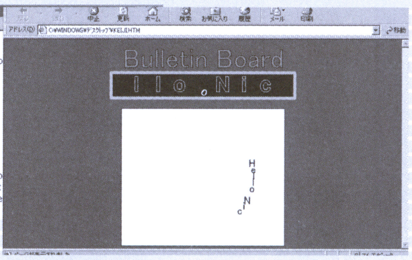
InternetExporlerがあれば，動作させることが可能です。VBScript+DirectX専用のソフトウェアなどはありません。

## 本誌プログラムの基本入力方法

① メモ帳などのテキストエディタから，掲載されているリストを入力します(写真1)。



▲写真1



▲写真2

※プログラムによっては，別途画像ファイルを必要とする場合があります。その場合は，誌面に掲載されている画像写真を参考に，ペイントなどのソフトを用いて画像ファイルを作成します。

②リストを入力後，入力したファイルの拡張子を.htmもしくは.htmlとして保存し，InternetExporlerで入力したファイルを読み込ませれば実行できます(写真2)。

# Excel(VBA)

## 製品紹介

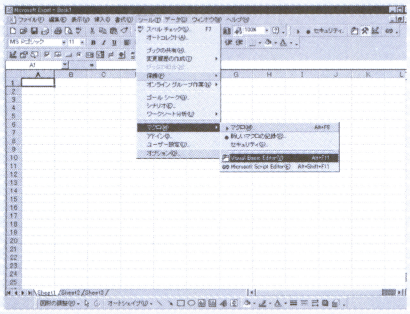
Excel(VBA)とは，MicrosoftのExcelで動作するVisual Basic for Applications(=VBA)のことを示します。Excelがあれば，特別なソフトウェアなしで，プログラミングを行うことで可能です。

## 入手方法

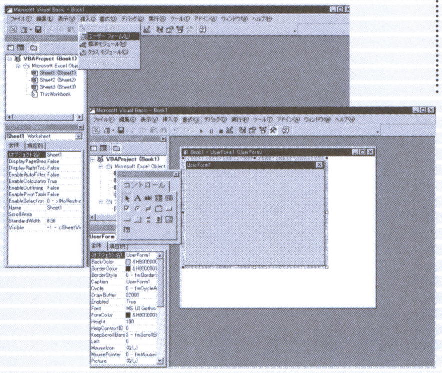
Microsoft Excelは，パソコンショップなどで購入可能です。パソコンによっては，最初からプレインストールされている場合もあります。

## 本誌プログラムの基本入力方法

①EXCELを起動し，メニューの[ツール][マクロ][Visual Basic Editor]を選択します(写真1)。

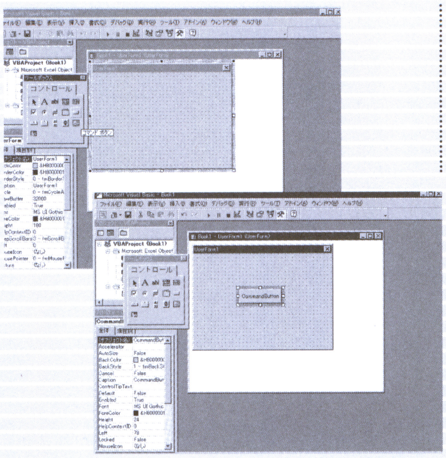


▲写真1

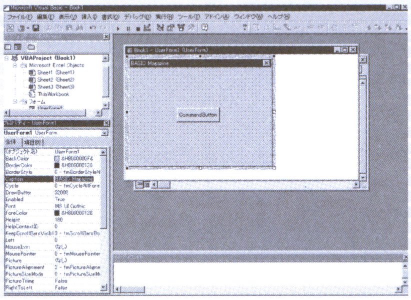


▲写真2

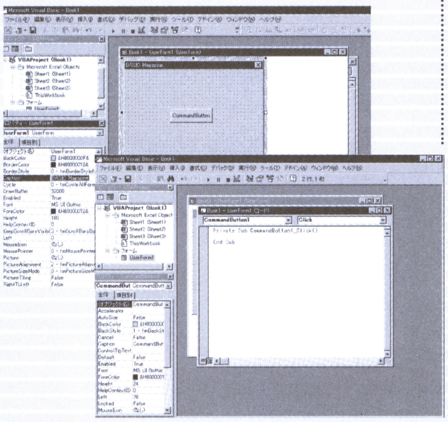
②[挿入]メニューから[ユーザーフォーム]を選択します(写真2)。
③ツールボックスのコントロールをクリックしたあと，UserForm1をクリックし，コントロールをUserForm1に貼り付けます(写真3)。
④画面のデザイン(コントロールの貼り付け)が終わったら，プロパティウインド



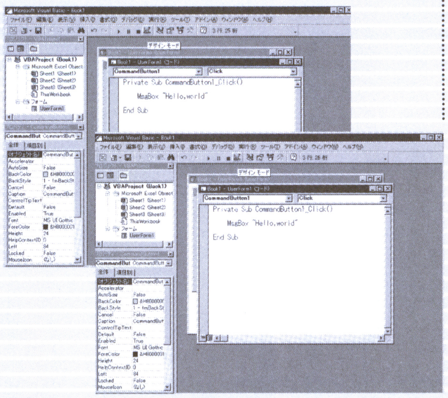
▲写真3



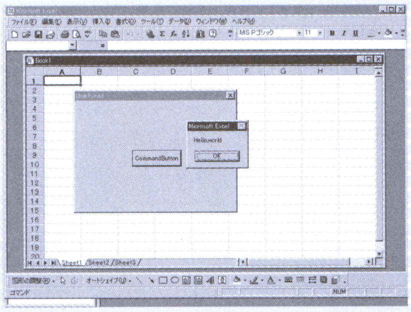
▲写真4



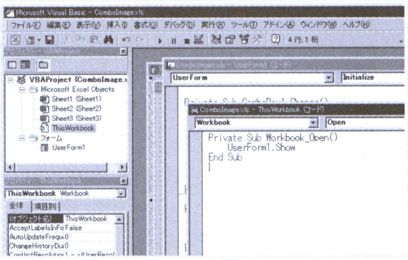
▲写真5



▲写真6



▲写真7



▲写真8

ウでプロパティの設定を行います(写真4)。
⑤プロジェクトウインドウのUserForm1を右クリックしコードの表示を選ぶ(もしくはコントロールをダブルクリック，コントロールを右クリック後，[コードの表示]を選ぶ)と，プログラムを入力するコードウインドウが表示されます(写真5)。
⑥表示されたコードウインドウにリストを入力します。コントロールとイベントがわかっている場合には，コントロールとイベントを選択して，適切なイベントにリストを入力します(写真6)。
⑦ [実行]メニューから[sub/ユーザフォームの実行]を選びます。または，メニューの下の三角のボタンをクリックしたり，F5キーを押してもプログラムを実行することができます(写真7)。
⑧WorkbookのOpenイベントで開きたいフォームのShowメソッドを実行すると，Excelブックを開くと同時にプログラムを実行することができます(写真8)。

※プログラムによって，ここで紹介した以外の作業を行う場合があります。その場合は，掲載されているプログラムの入力方法の項目にしたがって入力作業を行ってください。

# Javaアプレット

## 製品紹介

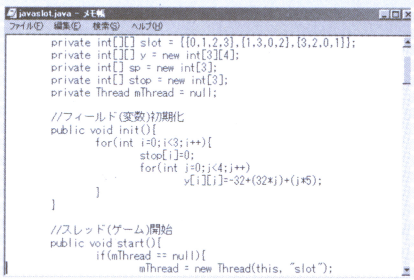
Javaアプレットとは，Java言語で作られたWebブラウザ上で動作するプログラムです。プログラムの制作には，JDKというJava専用の開発ソフトウェアが必要になります。現在のJDKは，Version1.3が主流となっています。

## 入手方法

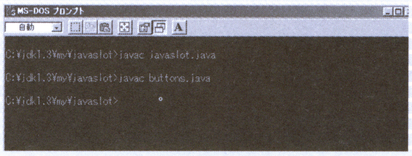
JDKは，サンマイクロシステムズのホームページから入手できます。また，市販されているJavaの入門書やJavaを扱っている雑誌のCD-ROMに収録されているものもあります。

## 本誌プログラムの基本入力方法

① メモ帳などのテキストエディタから，掲載されているリストを入力します(写真1)。Javaアプレットのプログラムには，実行させるためのHTMLファイルも必要になるので，掲載されているHTMLファイルも合わせて入力します。



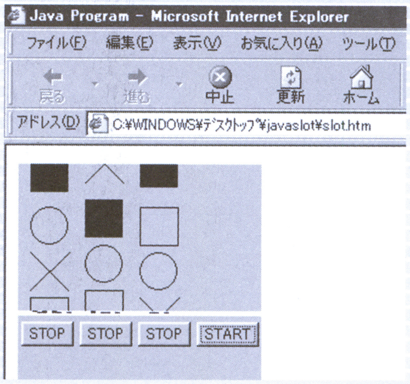

▲写真1




▲写真2

※プログラムによっては，別途画像ファイルを必要とする場合があります。その場合は，誌面に掲載されている画像写真を参考に，ペイントなどのソフトを用いて画像ファイルを作成します。

② リストを入力後，JDKのコンパイラを使用して，入力したリスト(ソースファイル)をコンパイルします (写真2)。




▲写真3

③ コンパイル後，InternetExplorerなどのWebブラウザソフトに，入力したHTMLファイルを読み込ませればプログラムが動作します(写真3)。その際，コンパイル時に生成されたclassファイルとHTMLファイルを同じフォルダ内に置いておく必要があります。また，JDKのappletviewerというソフトでも，実行させることが可能です。

# PIECE

## 製品紹介

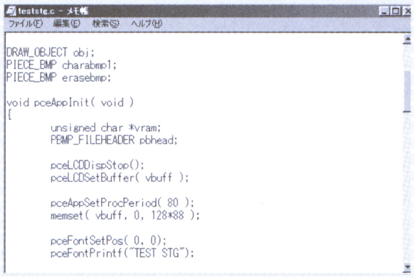
PIECEとは，アクアプラスから発売されている PC用携帯ゲーム機です。付属しているC言語開発ツールで，PIECEで動作するプログラムの制作が可能となっています。

## 入手方法

パソコンソフト店で購入可能です。価格は9800円(税別)となっています。取り扱い店舗は，ＰＩＥＣＥオフィシャルWebPage(http://www.piece-me.com/)でご確認ください。

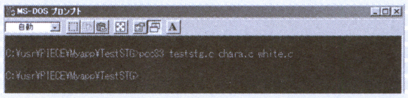
## 本誌プログラムの基本入力方法

① メモ帳などのテキストエディタから，掲載されているリストを入力します(写真1)。




▲写真1

② リストを入力後，PIECE専用のコンパイラを使用して，入力したリスト(ソースファイル)をコンパイルします (写真2)。




▲写真2

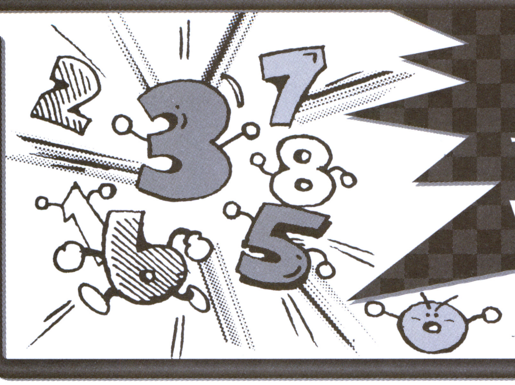
③ コンパイル後，PIECE本体とPCをつないだ状態で，コマンドラインから“Run ファイル名”(.srfファイルの名前)と入力すれば，PIECE上でプログラムが動作します(写真3)。

▲写真3

また，PIECEに付属しているコミュニケータや転送コマンドを使用して，ＰＩECE本体にプログラム転送した後，実行する方法もあります。詳しい方法は，ＰＩECE本体に付属しているマニュアルを参照してください。

※ PIECEについては，本誌2002年１月号，２月号にて特集していますので，より詳しい制作方法を知りたい方は，そちらも合わせてご覧ください。

# Windows(Visual Basic V6.0)用



レトロな雰囲気を味わえる

# 攻(せ)めてくる数字軍団(すうじぐんだん)

有田　安彦(32歳)

入力方法は46ページをご覧ください。

## 内容

20年ほど前のカシオの電卓に付いていたまぼろしのゲームです。いたってシンプルで，時間つぶしに最適なゲームです。

## 遊び方

ゲームを起動後，画面左横の「Game Start」ボタンを押すとゲームが開始します(**写真1**)。

左側からランダムに数字が攻めてくるので，画面上の「0」ボタンもしくは，テンキーの 0 を押して右側の数字を変更して，攻めてくる数字と同じ数にし，画面上の「.」ボタンもしくは，テンキーの . を押して，攻めてくる数字を消していく…とまあ，簡単なゲームです(**写真2，3**)。

テンキーの 0 もしくは，画面上の「0」ボタンを押した時の数字は，1，2，3，4，5，6，7，8，9，nと変化します。

数字が自分のところまで攻めてくると，残機(自分の数字の前の棒で表示)が1減ります。3回攻められるとゲームオーバーです(**写真4**)。

## プログラムの入力方法

①まず**フォーム**の新規作成を行います。

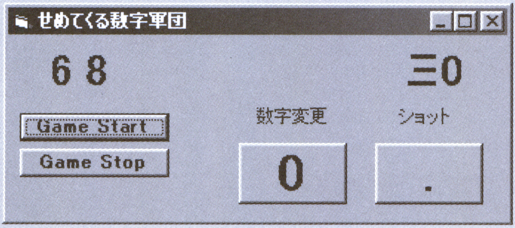
②[Lbl_no」という数字表示用のラベルを配列で，左から12個並べます。このとき左からINDEX番号を0～11と設定します。最初にCaptionに数字を書き，フォント調節したものをコピーして配置していくと便利です。

③数字の左下に「Bu_start」というゲームスタート用のボタンを配置します

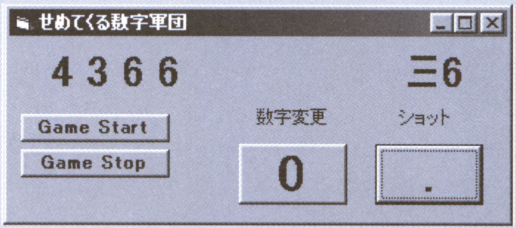
④その下に「Bu_stop]というゲーム終了用のボタンを配置します。

⑤その右に「Bu_key」という配列ボタンを作成し「0」とCaptionに書き，フォントを大きくします。このボタンをコピー＆ペーストで配置して配列にし，配置したボタンのCaptionを「.」と変更します。

⑥次にタイマーを設定します。名前を

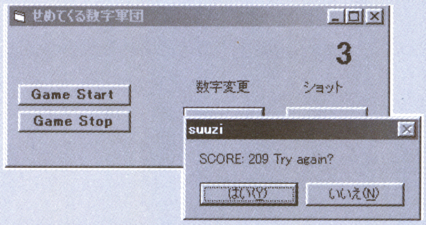



《写真1》なつかしの電卓ゲームをVBで楽しもう




《写真2》攻めてくる数字と同じ数字を発射

《写真3》素早く消さないと，数字に攻められてしまう




《写真4》昔遊んだ人なら，簡単に高得点を出せるかも？

《OVER FLOW》みなさん，新生活にはもう慣れましたか？新しい出会いはありましたか？このコーナーは，身近な話題，出来事や本誌の感想，意見など，なんでもアリ！「とじ込みハガキ」か，ホームページの「Internet読者ハガキ」でどしどし参加してください。

## 《第１表》プロパティ表

| | |
|---|---|
| フォームの設定 | |
| Name | Form1 |
| Caption | せめてくる数字軍団 |
| Height | 2250 |
| Left | 105 |
| KeyPreview | -1 'True |
| LinkTopic | Form1 |
| ScaleHeight | 1845 |
| ScaleWidth | 5055 |
| StartUpPosition | 3 'Windows の既定値 |
| Top | 105 |
| Width | 5175 |
| コマンドボタンの設定 | |
| Name | Bu_start |
| Caption | Game Start |
| Font | ＭＳ Ｐゴシック |
| …Size | 10 |
| …ｽﾀｲﾙ | 太字 |
| Height | 285 |
| Left | 120 |
| TabIndex | 4 |
| Top | 780 |
| Width | 1515 |
| Name | Bu_stop |
| Caption | Game Stop |
| Font | ＭＳ Ｐゴシック |
| …Size | 10 |
| …ｽﾀｲﾙ | 太字 |
| Height | 285 |
| Left | 120 |
| TabIndex | 5 |
| Top | 1140 |
| Width | 1515 |
| Name | Bu_key(0) |
| Caption | 0 |
| Font | ＭＳ Ｐゴシック |
| …Size | 26 |
| …ｽﾀｲﾙ | 太字 |
| Height | 615 |
| Index | 0 |
| Left | 2340 |
| TabIndex | 1 |
| Top | 1080 |
| Width | 1095 |
| Name | Bu_key(1) |

| | |
|---|---|
| Caption | . |
| Font | ＭＳ Ｐゴシック |
| …Size | 26 |
| …ｽﾀｲﾙ | 太字 |
| Height | 615 |
| Index | 1 |
| Left | 3720 |
| TabIndex | 0 |
| Top | 1080 |
| Width | 1095 |
| Name | Lbl_no(0〜11) |
| Caption | 0 |
| Font | ＭＳ Ｐゴシック |
| …Size | 22 |
| …ｽﾀｲﾙ | 太字 |
| Height | 495 |
| Top | 120 |
| Width | 315 |
| Name | Lbl_no(0) |
| Index | 0 |
| Left | 420 |
| TabIndex | 6 |
| Name | Lbl_no(1) |
| Index | 1 |
| Left | 780 |
| TabIndex | 7 |
| Name | Lbl_no(2) |
| Index | 2 |
| Left | 1140 |
| TabIndex | 8 |
| Name | Lbl_no(3) |
| Index | 3 |
| Left | 1500 |
| TabIndex | 9 |
| Name | Lbl_no(4) |
| Index | 4 |
| Left | 1860 |
| TabIndex | 10 |
| Name | Lbl_no(5) |
| Index | 5 |
| Left | 2220 |
| TabIndex | 11 |
| Name | Lbl_no(6) |
| Index | 6 |
| Left | 2580 |

| | |
|---|---|
| TabIndex | 12 |
| Name | Lbl_no(7) |
| Index | 7 |
| Left | 2940 |
| TabIndex | 13 |
| Name | Lbl_no(8) |
| Index | 8 |
| Left | 3300 |
| TabIndex | 14 |
| Name | Lbl_no(9) |
| Index | 9 |
| Left | 3660 |
| TabIndex | 15 |
| Name | Lbl_no(10) |
| Index | 10 |
| Left | 4020 |
| TabIndex | 16 |
| Name | Lbl_no(11) |
| Index | 11 |
| Left | 4380 |
| TabIndex | 17 |
| ラベルの設定 | |
| Name | Label1(0) |
| Caption | 数字変更 |
| Height | 255 |
| Index | 0 |
| Left | 2520 |
| TabIndex | 2 |
| Top | 720 |
| Width | 855 |
| Name | Label1(1) |
| Caption | ショット |
| Height | 195 |
| Index | 1 |
| Left | 3960 |
| TabIndex | 3 |
| Top | 720 |
| Width | 615 |
| タイマーの設定 | |
| Name | Ti_space |
| Interval | 2000 |
| Left | 1740 |
| Top | 900 |

「Ti_space」としてEnabledを「True」に設定します。

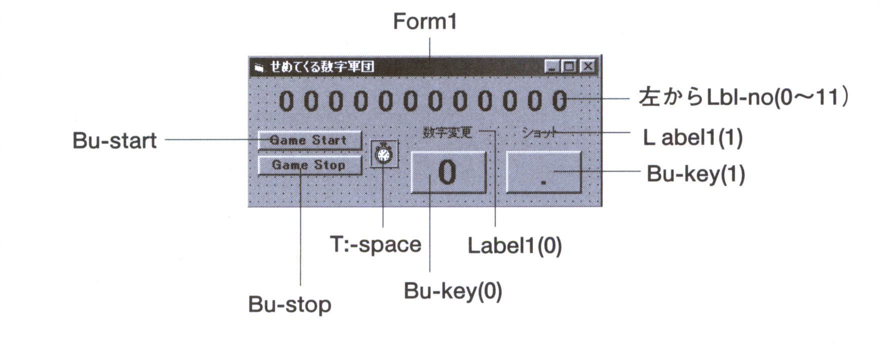
⑦あとはソースコード(**リスト１**)を書いて，できあがりです。

## 改良のアドバイス

画面構成をラベル表示からピクチャー表示に変更したりとか，数字キーをもっとたくさんつけて電卓機能をつけるなど…いろんな変更方法があると思います。

表示を大きくしても，結構たのしめます。桁数を増やして「攻めてくる英語軍団」とかにすると，おもしろいかも。あと，効果音とかも追加すると楽しいと思います。

カシオの電卓のときは「数字の合計が10になるとＵＦＯが出現する」構造だったと思いますが，今回のゲームでは，再現していません。




《第1図》フォームの配置図

## CHECKER FLAG

**影**：なつかしい形式のゲームだな〜。今の読者の中には，電卓にゲームが付いていたことを知らない人もいそうだね。

**編**：テンキーで遊ぶと，実際の電卓で遊んでいるような雰囲気を味わえますね。

**Dr.D**：ワシも懐かしみながら遊んだが，けっこう気になる点が多かった。まずは数字の消える反応が遅いという点じゃな。数字が移動したときに再描画するのではなく，ショットボタンと連動させて数字描画の更新を行えば，よりスピーディーなゲームになるはずじゃ。また，自分のところまで攻められたときに，何か演出がほしい。最低でもメッセージ表示は行ってほしいところじゃ。今のままでは攻められたという事がプレイヤーに伝わらないゾ。

**つぐ美**：リスト面では，各ルーチン内容のコメントが記述されている点がウレシイわね。

★学校訪問記，内容は良かったけど，私の名前，まちがえてます^^;。／まかべさん，PIECEの2月賞受賞おめでとうございます。（埼玉県熊谷市・**ク→チ17歳**）……【**編**：大変，失礼しました。ごめんなさい。**影**：学校から依頼されて送付したチラシは訂正しておきました。】

### 《リスト》攻めてくる数字軍団のプログラム・リスト

```
'***********************************************
'          攻めてくる数字軍団
'***********************************************
'TI_SPACE          数字の表示間隔タイマー
'Bu_kry(0)         テンキーの0
'      (1)         テンキーの.

Dim gi_you As Integer
Dim gl_score As Long
Dim gi_rest As Integer
Dim gi_ene(9) As Integer
Dim gi_digi(11) As Integer
Dim gi_pos As Integer
Dim gb_game As Boolean          '''true = play  false = stop
Dim gi_level As Integer
Dim gi_cnt As Integer
'''
'''


Private Sub Bu_key_Click(Index As Integer)

    If gb_game = True Then
        Select Case Index
        Case 0
            gi_you = gi_you + 1
            If gi_you > 10 Then
                gi_you = 1
            End If
            Sub_you
        Case 1
            '表示のチェック
            Sub_check
        End Select
    End If


End Sub

Private Sub Bu_start_Click()
    '''ゲームの開始
    Dim l_ans As Integer

    l_ans = MsgBox("ゲームを開始しますか?", vbYesNo)
    If l_ans = vbYes Then
        Sub_start
    Else
        gb_game = False
    End If

End Sub

Private Sub Bu_stop_Click()
    '''ゲームの終了
    gb_game = False
End Sub


Private Sub Form_Load()
'===============================================
'   開始の初期設定
'===============================================
    Dim l_loop As Integer

    'ゲーム用パラメーターの初期設定
    For l_loop = 0 To 9
        gi_ene(l_loop) = 0
    Next l_loop

    'デジタル表示状態の初期化
    Sub_setclr


End Sub

Sub Sub_game()
'===============================================
'   表示
'===============================================
    Dim l_loop As Integer
    For l_loop = 0 To 9
        Select Case gi_ene(l_loop)
        Case 1, 2, 3, 4, 5, 6, 7, 8, 9
            Lbl_no(l_loop).Caption = Trim(Str(gi_ene(l_loop)))
        Case 0
            Lbl_no(l_loop).Caption = " "
        Case 10
            Lbl_no(l_loop).Caption = "n"
    End Select
    Next l_loop
End Sub

'===============================================
```

★母校には花見に良い場所があるので，後輩をまじえて部室で宴会です。でも，桜もちとお茶のみですよ！（愛知県知多郡・**四谷一獲26歳**）……【**つぐ美**：さあ，お花見に行きますよ！**編**：あれ？影さんがいないよ。**つぐ美**：影さんは場所取りに朝一番で出発しました。】

```
'   キー処理
'=============================================
Private Sub Form_KeyPress(KeyAscii As Integer)
    If gb_game = True Then
        Select Case Chr(KeyAscii)
        Case "0"
            gi_you = gi_you + 1
            If gi_you > 10 Then
                gi_you = 1
            End If
            Sub_you
        Case "."
            '表示のチェック
            Sub_check
        End Select
    End If
End Sub

Sub Sub_setclr()
'=============================================
'   表示のクリア
'=============================================
    Dim l_loop As Integer
    For l_loop = 0 To 9
        Lbl_no(l_loop).Caption = " "
    Next l_loop
End Sub

Sub Sub_eneclr()
'=============================================
'   デジタル表示情報のクリアー
'=============================================
    Dim l_loop As Integer
    For l_loop = 0 To 9
        gi_ene(l_loop) = 0
    Next l_loop
End Sub

Private Sub Ti_space_Timer()
'=============================================
'   タイマー処理（敵の数字の表示処理）
'=============================================
    If gb_game = True Then
        Sub_ene
    End If
End Sub
```

```
Private Sub Sub_ene()
'=============================================
'   残り表示
'=============================================
    Dim l_loop As Integer
    Dim li_ans As Integer

    gi_pos = gi_pos + 1
    If gi_pos > 9 Then
        gi_rest = gi_rest - 1
        Sub_eneclr
        Sub_setclr
        Sub_rest
        Sub_you
        gi_pos = 0
        If gi_rest < 1 Then
            l_ans = MsgBox("SCORE:" & Str(gl_score) & " Try again?", vbYesNo)
            If l_ans = vbYes Then
                Sub_start
            Else
                gb_game = False
            End If
        End If

    End If

    For l_loop = 8 To 0 Step -1
        gi_ene(l_loop + 1) = gi_ene(l_loop)
    Next l_loop

    gi_ene(0) = Int(Rnd(1) * 10) + 1
    Sub_game

End Sub

'=============================================
'   残り表示
'=============================================
Sub Sub_rest()
    Select Case gi_rest
    Case 1
        Lbl_no(10).Caption = "一"
    Case 2
        Lbl_no(10).Caption = "二"
    Case 3
        Lbl_no(10).Caption = "三"
    Case Else
```

★CD-Rの特集をやってみてください。実際どれがいいのかわかりません。（福島市・**羽曽部健16歳**）……【**編集長**：そうだな，DVD-Rも含めて考えてみようか。**編**：学校の先生からも，裏技的な使いかたなど，授業では教えられない記事を求む声が上がってます。】

```
        Lbl_no(10).Caption = " "
    End Select
End Sub

'==============================================
'    ゲーム開始初期設定
'==============================================
Sub Sub_start()
    gb_game = True
    Sub_eneclr
    gi_rest = 3
    gi_pos = 0
    gl_score = 0
    gi_level = 1
    gi_cnt = 0
    Sub_rest
    Sub_speed

End Sub

Sub Sub_speed()
    Dim l_int As Integer

    Select Case gi_level
    Case 1
        l_int = 2000
    Case 2
        l_int = 1500
    Case 3
        l_int = 1000
    Case 4
        l_int = 700
    Case 5
        l_int = 500
    Case 6
        l_int = 300
    Case 7
        l_int = 200
    Case Else
        l_int = 100
    End Select

    Ti_space.Interval = l_int
End Sub

'==============================================
'    データチェック
'==============================================
Sub Sub_check()
    Dim l_loop1 As Integer
    Dim l_loop2 As Integer

    For l_loop1 = 8 To 0 Step -1
        If gi_ene(l_loop1) = gi_you Then
            gi_cnt = gi_cnt + 1
            If gi_cnt > 10 Then
                gi_cnt = 0
                gi_level = gi_level + 1
                Sub_speed
            End If
            For l_loop2 = l_loop1 + 1 To 9
                gi_ene(l_loop2 - 1) = gi_ene(l_loop2)
            Next l_loop2
            gi_pos = gi_pos - 1
            gl_score = gl_score + gi_ene(I)
            Exit Sub
        End If
    Next l_loop1
    Sub_ene
    Sub_game

End Sub

Sub Sub_you()
'==============================================
'    タイマー処理（敵の数字の表示処理）
'==============================================

    Select Case gi_you
    Case 1, 2, 3, 4, 5, 6, 7, 8, 9
        Lbl_no(11).Caption = gi_you
    Case 10
        Lbl_no(11).Caption = "n"
    End Select

End Sub
```

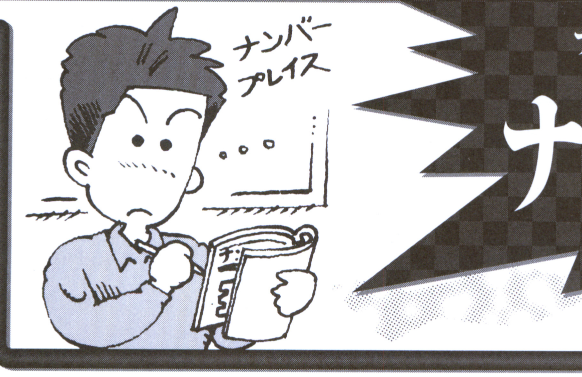
★とても勉強になっていいです。（新潟県長岡市・**南　寿和20歳**）……【**影**：このお便り，簡潔でいいです。**編**：どのへんが勉強になっているのか具体的に書いてほしいですよね。**影**：どことは特定できないんでしょ。**編**：ま，我々の励みになることは確かです。】

**超大作プログラム** リストが誌面に掲載できない作品です。
BMホームページ(http://www.basicmagazine.net)からダウンロードしてお楽しみください。

# Windows(Visual Basic V6.0)用

ナンバープレイス解答は，これで完璧？

# ナンバープレイっす




米久保 直樹(18歳)

## はじめに

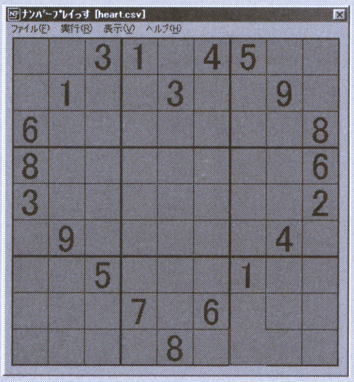
ナンバープレイスを解くためのソフトです。ナンバープレイスとは，９×９の81個のマスの中に，縦にも横にも，３×３のマスの中にも，それぞれ１～９の数字が一つずつ入るように数字を入れるパズルです。

## 使用方法

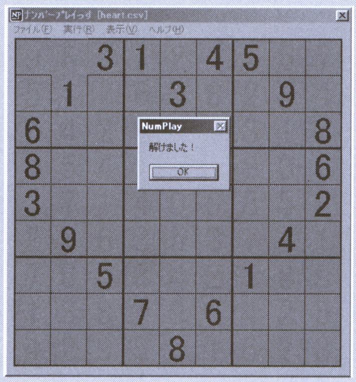
問題を入力して，メニューの“解析”を実行すれば，回答が赤い文字で表示されます(写真１，２)。

“矛盾の調査”は「この問題，本当に解けるのかよ」という人のための機能で，回答を表示せず，問題が解けるかどうかだけ表示します(写真３)。

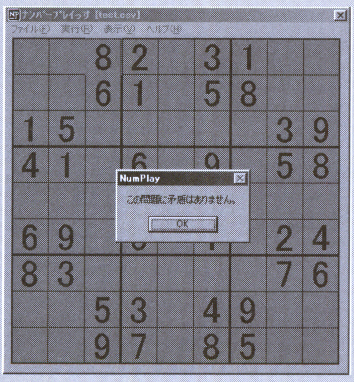
“回答の消去”は，赤い文字で表示された回答だけを消去します。



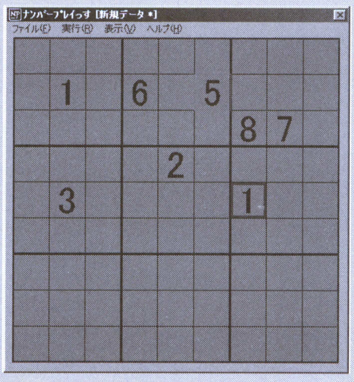

《写真１》ナンバープレイスの問題を瞬時に表示




《写真２》解答はこんな感じで表示させる




《写真３》問題が解けるかのチェックも可能




《写真４》画面上で問題の入力ができるのも特徴

## 問題の入力方法

ファイルから読み込む方法と，直接入力する方法があります。

ファイルは，カンマ区切りのcsvファイルです。これはメモ帳やExcelなどで作成できます。

直接入力するには，まず値を入力したいマスをクリックします。すると，青い枠が出ますから，そこで数字を入力してください。値を消すには，BackSpaceキーを押してください。また，青い枠はカーソルキーで移動させることができます(写真４)。

## プログラムの内容

このプログラムは，消去法で問題を解いています。つまり，マスごとに絶対に入ることの無い数字をリストアップしていき，中に入る数字が一つに限られたときに，そのマスの数字を決定します。

ただし，この方法では回答が複数ある問題に対応できないので，代入法も合わせて利用しています。代入法とは，マスの中にとりあえず可能性のある数字を入れてみる方法です。ちなみに代入法はメニューで利用するかしないか変更できます。

## 苦労した点

はじめは，Visual C++でコマンドラインプログラムとして作成したものを，使

★付録がつくのは良いけど，いきなり値段が2倍以上になったらイヤだなぁと思いました。(島根県美濃郡・K24歳) ……【編：付録CDは必要か否かの論議が引き続き行われているようです。編集部でも年1回は付けてもいいのでは，など，意見がでているのですがあ…。】

いにくいといった理由からVBに移植したのですが，VBが遅すぎて苦労しました。Visual C++で1秒もかからない処理でも，VBでは20秒近くかかってしまいましたから…。

仕方ないので，問題解析のアルゴリズムを工夫して，なんとか実用的な速度までもっていきました。

## お礼・おわりに

このプログラムのアイコンは，母校の某工業高校の後輩が作ってくれたものです。この場を借りてお礼を言いたいと思います。市販の，とあるソフトにとてもよく似ているような気がしますが……。

このプログラムを作成しようと思ったのは，会社の同僚がやたらと難しいナンバープレイスの問題をもってきたことがきっかけです。

あまりにも難しかったので，本誌の2000年1月号に載っていたナンバープレイスソフトに解かせたところ，散々時間をかけたあげくに“解けません”といわれてしまい，「これは自作するしかないな」と思って作成しました。

3日もかけて結局解けなかった問題をあっさり1秒ほどで解かれて，うれしいような悲しいような……。

## CHECKER FLAG

**編**：以前に掲載した同じ内容のソフトで解けない問題でも解けるというのがいいですね。ソフト制作の動機がしっかりしているからか，作りもしっかりしています。

**Dr.D**：問題解答の強化だけでなく，問題作成や矛盾調査機能など，独自の機能を搭載している点に好感がもてるのぉ。やはり以前掲載されている投稿作品と同じアイデアの作品の場合は，機能や追加アイデア，もしくはリスト面で抜き出たものがほしいところじゃな。

**つぐ美**：ただ，機能を追加した分リストの量が増えてしまい，残念ながら超大作での掲載となってしまいましたね。



入力方法は46ページをご覧ください。

## Pick Up LIST

《リスト1》ファイル読み込み部分のリスト

```
' ファイルの読み込み
Private Function FileLoad()
    Dim i As Integer, j As Integer
    Dim tmp(8, 8) As Integer

    On Error GoTo ReadError

    Open path & filename For Input Access Read As #1

    For i = 0 To 8
        Input #1, tmp(i, 0), tmp(i, 1), tmp(i, 2), tmp(i, 3), tmp(i, 4) _
          , tmp(i, 5), tmp(i, 6), tmp(i, 7), tmp(i, 8)
    Next i

    Close #1

    For i = 0 To 8
        For j = 0 To 8
            If tmp(i, j) > 9 Or tmp(i, j) < 0 Then
                GoTo ReadError
            End If
        Next j
    Next i

    For i = 0 To 8
        For j = 0 To 8
            field(i, j) = tmp(i, j)
            answer(i, j) = False
        Next j
    Next i

    Call SetTitle(filename, False)

    Exit Function

ReadError:
    Close #1
    MsgBox "このファイルは正しく読み込めません。"
    Call SetPath("")
    Exit Function

End Function
```

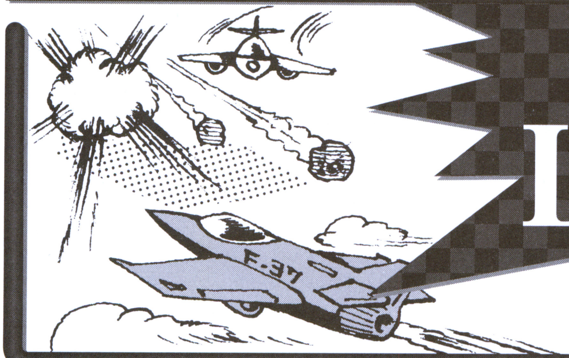
## Dr.Dのワンポイント解説

**Dr.D**：この手の作品では，やはり問題の解析ルーチンを紹介したいのじゃが，リスト量が多いため，ここではファイル処理のリストを掲載してみた。このルーチンは，csvファイルを読み込む処理を行うので，値を格納する配列(tmp(8,8)で定義されている)の大きさを変更すれば，汎用性のあるcsvファイル読み込みルーチンとしても使用できるのではないかな？

超大作プログラム

リストが誌面に掲載できない作品です。
BMホームページ(http://www.basicmagazine.net)からダウンロードしてお楽しみください。

# Windows(Visual Basic V6.0+DirectX)用



DirectXの定番投稿作品第４弾

# Laser Shoot Ⅳ（レーザーシュートフォー）

伊藤　隆朗（16歳）

## ゲーム内容

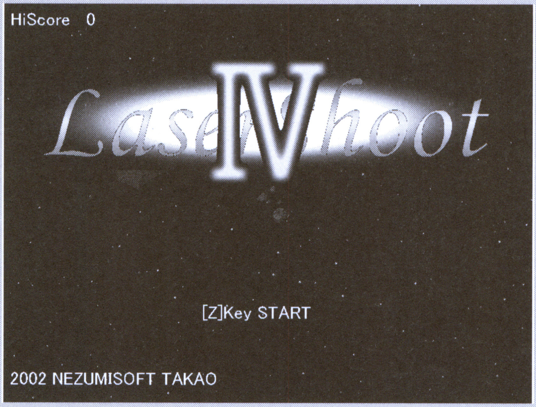
高校生になり，久しぶりにレーザーシュートの最新作を作りました。

今回も，内容は隕石と飛行機を,ただひたすら撃ち落としていくだけのゲームで，ストーリーも何もありません…。

前作(Ⅲ)との違いは，完全な擬似３Ｄのシューティングになっています。迫力は前作よりもアップしていると思います。さらに，登場する敵機３機はただ突っ込んでくるだけでなく，一応考えて動いています。

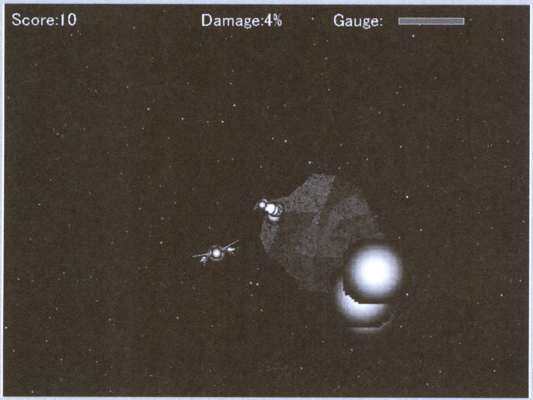
## 制作環境など

制作環境はセレロン433MHz，ビデオメモリ8M，メモリ256Mです。この環境で動作が結構きびしいです…。解像度は800×600，16ビットカラーです。

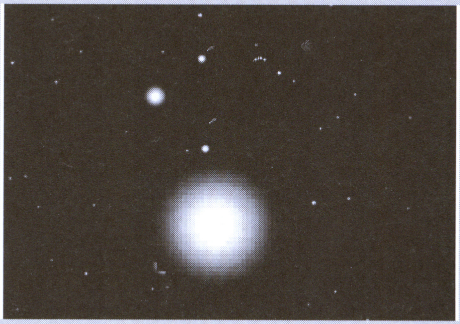



《写真１》人気投稿作品レーザーシュート最新作

《写真２》今回は，完全な3Dシューティング




《写真３》迫力ある画面がこの作品の特徴



《写真４》シールドなどゲーム面の工夫もある

★買い始めて10年目にして，はじめて売り切れを経験しました。他の市まで買いに行ったよ。（岩手県釜石市・鉄人坂井24歳）……
【編：最近，書店からの注文電話が増えたようですね。販売部長：バックナンバーの注文も多いんだよ。まとめて1年分なんていうのもね。】

画像はDirectDraw，BGMはDirectMusic，効果音はDirectSound(3DSound)，キー入力はDirectInput，半透明処理にDirect3Dを使っています。

半透明処理はグラフィックカードの性能が低いと，処理されない場合があります。それと，DirectX8以上がインストールされていると，BGMが流れません。

## 遊び方＆操作説明

ただ単純に，迫ってくる飛行機と隕石を，レーザーで撃って破壊してください（**写真２，３**）。画面中央にあるダメージが100%になるとゲームオーバーです。

タイトル画面で，Zキーを押すとゲーム開始です。タイトル画面で，カーソルキーの上下でゲームスピードの調節ができます(**写真１**)。数が少ないほどスピードが速くなります。パソコンの動作環境に合わせてください。

ゲームの操作方法は，以下のとおり。

・カーソルキーで自機を移動できます。

・Zキーでレーザーを発射できます。

・Xキーを押すと，「連射レーザー」→「拡散レーザー」→「集積レーザー」→「連射レーザー」と順番に変更されます。各レーザーは発射されるまでの時間が違います。

敵のレーザーや隕石を防ぐには，Cキーを押すことでシールドを張ることが出来ます。Cキーを押している間，自機の周りにはシールドが張られます(**写真４**)。しかし，張っている間は画面上の右端のゲージが減っていき，なくなるとシールドは解除されてしまいます。ゲージは隕石を撃つことで少しずつ回復します。

３種類のレーザーとシールドを駆使して高得点を目指してください。

Escキーでゲームの強制終了ができます。

## 改造のアドバイス＆苦労した点

改良点は，ずばりゲーム内容です。ストーリーやステージをつけると立派なシューティングゲームになると思います。

3Dにするのは初めてなので結構苦労しました…。

## 作者のひとり言（次回予告？）

レーザーシュートシリーズも，Ⅳ作目…。次回作(Ⅴ)を作るとしたら，ゲームの基盤のようなものは固まったので，もっとステージとストーリーに力を入れるつもりです。今は，アクションRPGを作っているので，Ⅴを作るとしたらアクションRPGが完成してからでしょうか…。
何かありましたら，ホームページ：
http://up2.tok2.com/NEZUMI/
メールアドレス：ntakao@d4.dion.ne.jp
にでも意見をください。
次の投稿は，簡単なアクションゲームでも…。

### CHECKER FLAG

**つぐ美**：DirectX投稿作品では，人気がある伊藤クン作の『レーザーシュート』シリーズね。3Dならではの奥行きがあるので，迫力あるシューティングが楽しめます。

**影**：快適に遊ぶには，動作させるPCのスペックもチョット高い性能がほしいところだね。PCスペックによっては動作の遅さから，爽快感が感じられないかも？

**Dr.D**：辛口の意見を言えば，ゲーム内容が単調なのですぐに飽きがきてしまう感じじゃ。超大作なのでゲーム面での工夫もほしかったな。じゃが，プログラム作品としての完成度は素晴らしいものじゃ。この分なら次回のアクションゲームも期待できるのぉ。投稿を楽しみにしておるゾ。

## Pick Up LIST

《リスト１》オープニング処理の部分のリスト

```
'Ⅳの表示効果処理
For i = 0 To 50
WaitT
BltF 0, 0, 0, 0, 0, 800, 600, False
DStar
DMeteo
BltATF 0, 50, 100, 700, 150, 0, 0.5, 1
BltF 1, 50, 100, 0, 100, 700, 150, True
BltATF 1, 300, 75, 200, 200, i / 50, i / 50, i / 50
DDFlip
Next i

'テキスト表示効果
'SetColorの値を調整することで，フェードイン効果で表示している
For i = 0 To 17
WaitT
BltF 0, 0, 0, 0, 0, 800, 600, False
DStar
DMeteo
BltATF 0, 50, 100, 700, 150, 0, 0.5, 1
BltF 1, 50, 100, 0, 100, 700, 150, True
BltATF 1, 300, 75, 200, 200, 1, 1, 1
SetColor i * 15, i * 15, i * 15, i * 15, i * 15, i * 15
DText 10, 10, "HiScore  " & HSco
DText 300, 450, "[Z]Key START"
DText 10, 550, "2002 NEZUMISOFT TAKAO"
CopySur
DDFlip
Next i
```

### Dr.Dのワンポイント解説

**Dr.D**：このゲームを印象づけている要素の一つが，さまざまな演出を行っているオープニング画面じゃ。このオープニングは一つ一つの演出ごとに，掲載しているようなFor文ごとの記述がなされておるのじゃ。やや効率が悪いリスト記述ではあるのじゃが，各効果がどのように実現されておるかが，わかりやすいリストではある。試しに表示効果部分(このリストで言えばSetColor命令など)をコメント化して実行してみると良いじゃろう。

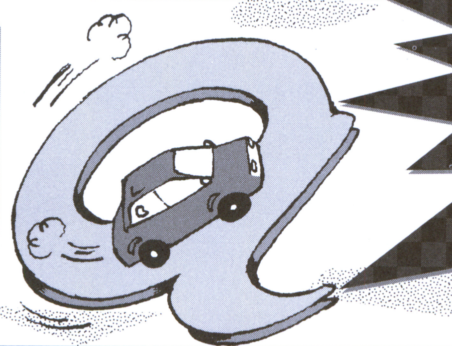


入力方法は48ページをご覧ください。

★マッキントッシュ用にオフィス（マイクロソフト）XPらしきものがあるのを思い出してしまった。デュアルCPUのあれは安くなるだろうか？（徳島県阿南市・T③29歳）……【**影**：アレね。アレはまだ安くならないんじゃないか？**つぐ美**：なんですか？**影**：秘密！】

(超大作プログラム) リストが誌面に掲載できない作品です。
BMホームページ(http://www.basicmagazine.net)からダウンロードしてお楽しみください。

# Windows(Visual Basic V6.0 + DirectX)用



アルファベットの３Ｄマップを疾走

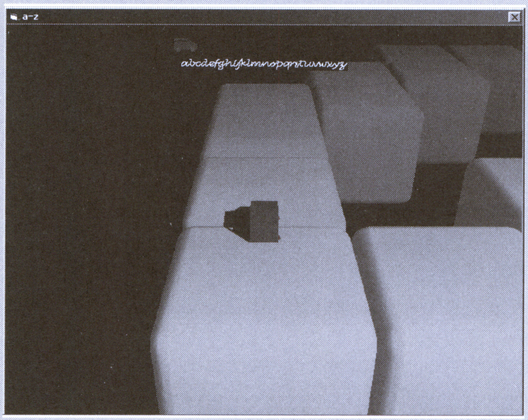
# a-z
（エー ゼット）

橘 一馬(30歳)

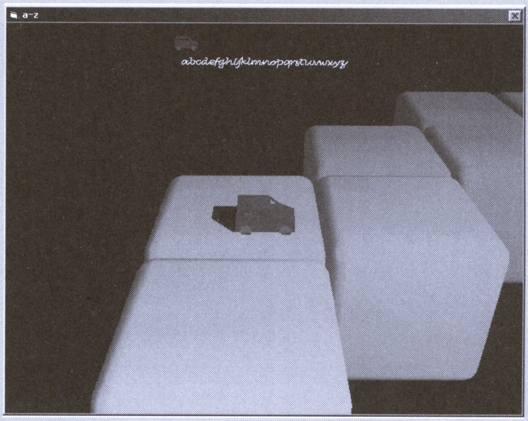
## 遊び方・操作方法

車を3Dマップに沿って操作していき，ずっと右奥に進んでいった所にある旗を取るのが，このゲームの目的です(写真１)。

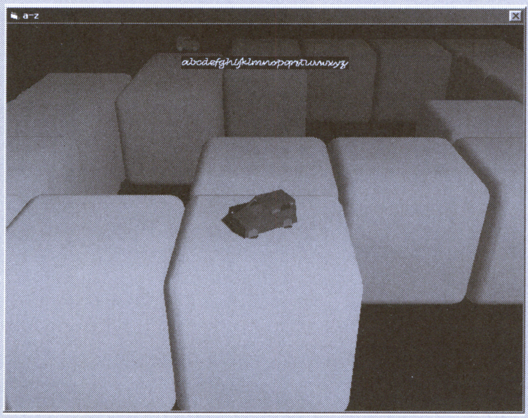
車は常に前へ進んでいきます。マウスを左クリックすると車が右回転するので，方向転換させながら進めていきます(写真２，３)。画面の上に表示してあるアルファベットの文字列(マップ)で現在の車



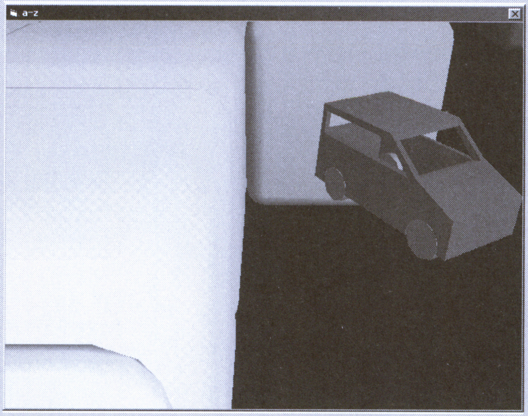
《写真１》ブロックの上を車が走り回る



《写真２》マップにそって方向転換していく



《写真３》右回転だけなので，地道な転回が必要



《写真４》マップから落ちてしまうとダメ

★ブロードバンド普及率ワースト1位の県に住んでいます。1.1％，約1/100。どうりで導入の強迫感がないわけだ。(福島市・南斗オレガ25歳)……【影：編集部内ブロードバンド普及調査の結果です。ADSL＝編集長，Dr.D。CATV＝編さん，つぐ美ちゃん。私は…内緒。】

の位置がわかります。車がマップから転落するとゲームオーバーです(写真4)。

## あれこれ

よく見るとマップ全体は，アルファベットのa-zの形なんです。わかりにくいと思いますが…。僕はcぐらいしか行ったことがありません。

Map.bmpを変えれば自由にマップが作れると思います。アイテムやトラップを作ったら，またおもしろくなるかな？

## おわりに

最近，ADSLにしました。とっても快適です。新しい使い方ができそうです。

**★参考にしたもの**

・効果音：音・辞典(TM) VOL.12［アメリカン・アニメ効果音／コメディ］

・Xファイル：Metasequoia

・プログラム：Visual Basicによる DirectX7活用術　川田徹，DirectX7 For Visual Basic　はじめるゲームプログラミング　藤田伸二

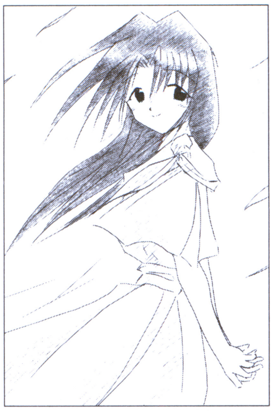
## CHECKER FLAG

**編**：車を動かすゲームではありますが，ドライブゲームというよりは，迷路ゲームといった方が正しいかもしれませんね。実際操作するのは，車の方向転換，それも右回りのみですから。

**つぐ美**：でも，編さんの言うように迷路ゲームとして認識すれば，けっこう面白いゲームなんじゃないかしら。ただ，マップどおりに走らせていくのは，けっこう難しいので，落ちた所から再開できるリプレイ機能が欲しかったところね。

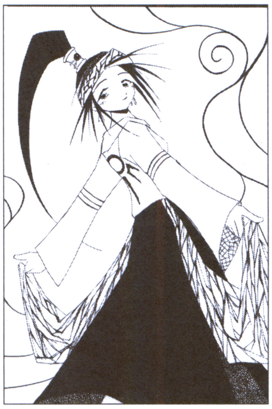
**Dr.D**：3D表示の車やマップだけでなく，車の影(シャドー)がついている点も3Dゲームらしいのぉ。ただ，こういった処理が重い機能がある分，快適に遊ぶには，それなりのスペックがあるPCが欲しいところじゃな。シャドーなどは，機能のON/OFFが選択できるようにしても良いかもしれんな。

**影**：車の直進スピードに加速処理がついていたりなど，チョットしたこだわりのある車の動きにも注目だゾ。



by 5吋2Dな皆様江

評…見つめないで♥(影)



by MINO様江

評…きまってル!!(編)



入力方法は48ページをご覧ください。

## Pick Up LIST

《リスト1》マップ作成部分のリスト

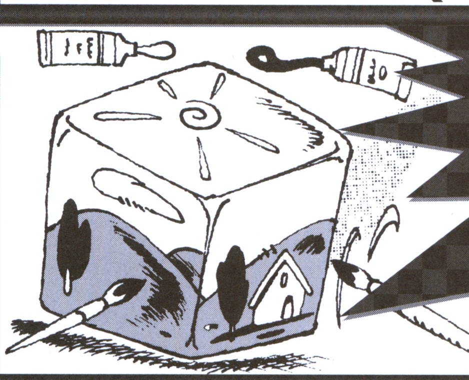
```
'マップ作成
    Form1.picMap.Visible = True
    Form1.Refresh
    For i = 0 To MAP_HEIGHT - 1
        For j = 0 To MAP_WIDTH - 1
            Select Case Form1.picMap.Point(j, i)
            Case vbWhite
                Map(j, i) = MAP_BLOCK
            Case Else
                Map(j, i) = MAP_NO
            End Select
        Next
    Next
```

## Dr.Dのワンポイント解説

**Dr.D**：説明文で，画像ファイルを変更すればマップが作れる，と書いてあることから推測できると思うが，このゲームは画像ファイルからマップデータを作成しておるのじゃ。画像ファイルの色をPoint命令で得て，その色が白だったらブロックを置く，違ったら何もない，という方法で作成を行っておる。処理内容は簡単じゃが，なかなかのアイデアと言えるじゃろう。

★はじめまして。来月号も楽しみにしています。さようなら。(宮城県仙台市・たけし16歳)……【編：ずいぶん，そっけないですね。影：初投稿ということなので，人見知りしてるんでしょ。そのうち「影さん，がんばって！」「影さん大好き！」なんて書いてくるさ。】

# Windows(VBScript + DirectX)用



キューブを使った3Dパズル

# キューブパズル

金井　義男（39歳）

## 内容

正立方体（キューブ）に6枚の画像が貼り付けられています。マウスクリックで，キューブを回転させて，画像を合わせるパズルゲームです。

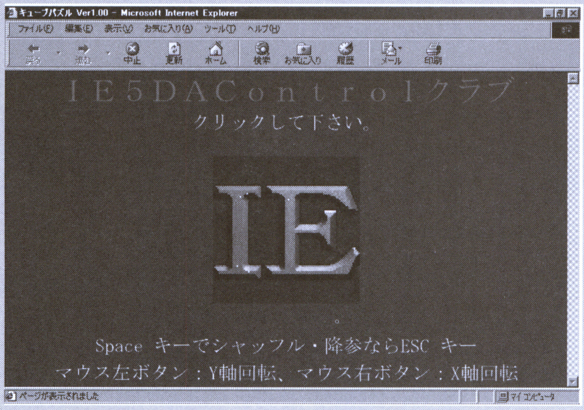
## 使い方・遊び方

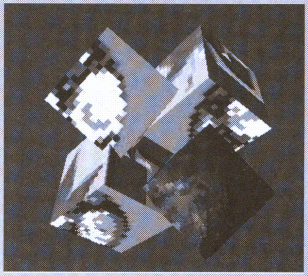
入力したリストの拡張子を.htmlにして，InternetExplorerで読み込ませて実行させます。その際，使用する画像ファイル(後述)は同じフォルダに置いてください。

画面の説明に従ってゲームを遊んでください。流れとしてはスペースキーで画像をシャッフルし，マウスのクリックでキューブを回転させて画像を揃えていきます。

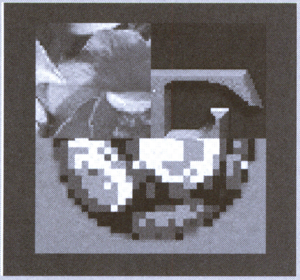
画像が揃うと，クリアメッセージボッ

《写真1》キューブ(立方体)の絵柄を揃える

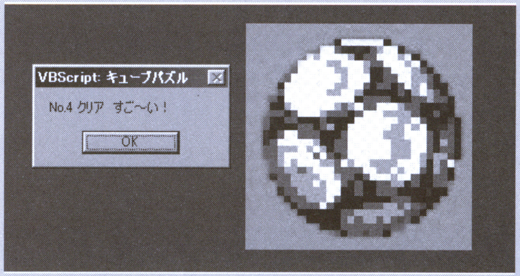


《写真2》まずはスペースキーで絵柄をシャッフル



《写真3》クリックしてキューブを回転させながら




《写真4》どれか一つ絵柄を揃えればOK

★（4月号48ページ）僕は「プログラム・チェックご苦労様」なんてことを書きますが…。やっぱ，白紙だと怖いし…。（長野市・サバ柄13歳）……【編：今月もプログラム・チェック大変でした。影さんなんか，他人のｉモードでｉアプリばっか，やってましたけど。】

クスが，表示されます。

使用する６枚の画像は，Image0～5.jpgと名前を付けて，htmlファイルと同じフォルダに保存すれば，好みの画像パズルができます。また，キューブの数は，ソースのGRID定数を変更すれば変えられますが，数が多くなるほど，処理が遅くなり，パワーのないPCでは，動かなくなります。画像サイズは，GRID数×50ピクセルが適切です。これは，ソースのSIZE定数で変更できます。

## 操作方法

画像をクリックすればキューブが回転します。マウス左ボタンでY軸回転，右ボタンでX軸回転しますが，X軸回転時にはY軸も回転してしまいます。この法則を理解できるまでは，苦労するでしょう。

Spaceキーを押すことで，シャッフルされます。降参ならESCキーを押すと，左上の表示画像でキューブを整列します。

## プログラムの入力方法

メモ帳を使って入力，改造ができますが，デバッガとテキストエディタ機能のあるツールを使用することをお薦めします。GRID数を増やす時には，セーブをしてから，実行してください。PCが動かなくなると困るので。

## 苦労した点・おわりに

キューブが回転した時に，入れ替える配列順序を考えるのに苦労しました。あと，3Dアニメーションの実行速度が遅いし，なめらかさもないのでテスト実行の待ち時間が長かったです。

本誌ではボツになったものや，バージョンUPしたものを私のホームページでも紹介していますので，暇のある方は見に来てください。

http://www.d7.dion.ne.jp/~kanaisun

## CHECKER FLAG

**つぐ美**：こういったパズルは，実物では似たようなものがけっこう存在しますが，パソコン上のゲームとしては珍しいかもしれませんね。

**影**：使用する画像ファイルを，好きな画像にして遊ぶのがオススメ。

**Dr.D**：ゲーム面では，せっかく3D処理をしているのだから，視点変更機能を搭載するなどして，もう少し3Dらしさを強調してほしかったところじゃな。リストもチト長めじゃが，ゲームとしてしっかりと形にしてある点は評価できるゾ。

**編**：3Dゲームの入門として，このリストを参照するのもいいかもしれません。DirectAnimationではありますが，3Dの基本的な処理はどの言語でも共通の知識ですからね。

画像ファイル例

VBScript +DirectX

入力方法は48ページをご覧ください。

《リスト1》html自動生成プログラム・リスト

```
<html>
<head>
<title> キューブパズル Ver1.00 </title>
</head>
<body BGCOLOR=PURPLE>
<SCRIPT LANGUAGE="VBScript">
'―――――――――――――――――――――――――――――――――――――――
'// グローバル変数の宣言
'―――――――――――――――――――――――――――――――――――――――
Option Explicit
Dim m              'As DAStatics
Dim mesh(5)        'As DAGeometry      '// 正方形の面（6枚）
Dim tximg(5)       'As DAImage         '// マッピング（テクスチャ貼りつけ画像6枚）
Dim block()        'As DAGeometry      '// 正方形立方体
Dim xang()'As Single'// X軸回転角度記録変数（ラジアン）
Dim yang()'As Single'// Y軸回転角度記録変数（ラジアン）
Dim rotx()'As DANumber          '// X軸角度変動値（定数SPEED秒で90度回転）
Dim roty()'As DANumber          '// Y軸角度変動値（定数SPEED秒で90度回転）
Dim group'As DAGeometry         '// 立方体の集合体
Dim messimg        'As DAImage         '// メッセージ イメージ
Dim ch             'As DANumber        '// クライアント ウィンドウの高さ
Dim snd            'As DASound         '// BGMサウンドイメージ
Dim snd1  'As DASound           '// 無音サウンドイメージ
Dim snd2  'As DASound           '// キューブ回転サウンドイメージ
Dim cx             'As Long        '// マウスダウン時コントロール内のX座標値
Dim cy             'As Long        '// マウスダウン時コントロール内のY座標値
Dim number()       'As Long        '// 面番号記録用変数配列
Dim poschg    'As Variant        '// 軸位置による配列番号置き換え順序代入用配列
Dim last  'As Long  '// キューブの数(縦×横)
Const PI = 3.14159265359        '// 円周率
Const SPEED = 0.3               '// キューブの回転時間(秒)
Const GRID = 2                         '// 縦横のキューブの数
Const SIZE = 50                        '// 分割画像サイズ(ピクセル)

'―――――――――――――――――――――――――――――――――――――――
'// 初期化処理
'―――――――――――――――――――――――――――――――――――――――
Sub Window_OnLoad()
'// 変数の宣言
Dim file  'As String            '// 画像パス文字列
Dim light 'As DAGeometry               '// ライト
Dim geo            'As DAGeometry             '// ジオメトリ
Dim camera'As DACamera                 '// カメラ
Dim geoimg'As DAImage                  '// レンダリング イメージ
Dim finalimg       'As DAGeometry             '// 最終 イメージ
Dim i              'As Long

  '// メートルライブラリを使用（右手座標系）
  Set m = DAControl.MeterLibrary
  For i = 0 To 5
    '// イメージの画像を読み込み（6枚の画像、GRID×50ピクセルが適正サイズ）
    file = "image" & i & ".jpg"
    Set tximg(i) = m.ImportImage(file)
  Next
  '// 軸位置による配列番号置き換え順序を Array:配列形式で値を代入
  poschg = Array(5341, 5042, 5143, 5240, 3210)
```

```
  '// Shapes DirectX 3Dトランスフォームで物体面の作成
  Call Meshmake()
  '// メッセージ文字イメージの作成
  Call Dispmesg()
  '// 光源(照明)の定義(アンビエン・ライト  白色)
  Set light = m.AmbientLight.LightColor(m.ColorRgb(1, 1, 1))
  Set geo = m.UnionGeometry(light, group)
  Set geo = m.ModifiableBehavior(geo)
  '// パースペクティブ カメラの定義
  Set camera = m.PerspectiveCamera(8, 7)
  '// 視野を広げて物体を小さく見せる。物体を小さくするより遠近感を出すテクニック
  Set camera = camera.Transform(m.Scale3(85, 85, 1))
  '// レンダリング = 3Dから2Dへ計算投影(オブジェクト)
  Set geoimg = geo.Render(camera)
  '―――――――――――――――――
  '// トータルイメージの作成
  '―――――――――――――――――
  '// Array関数は、配列形式のバリアント型変数を返します。(配列は,絵の重なる最上位から順に)
  Set finalimg = m.OverlayArray(Array(messimg, geoimg))
  '// コントロールにオブジェクトをセット
  DAControl.Image = finalimg
  '// 無音のDASoundオブジェクト
  Set snd1 = m.Silence
  '// キューブ回転音用サウンドイメージの読み込み
  Set snd2 = m.ImportSound("Brdsqu1.mp3").Sound
  Set snd = CreateObject("DirectAnimation.DASound")
  snd.Init m.Until(snd1, m.LeftButtonDown, m.Until(snd2, m.Timer(0.5), snd))
  '―――――――――――――――――
  '// アニメーション開始
  '―――――――――――――――――
  DAControl.Start                    '// コントロールにオブジェクトをセット
  DAControl.Sound = snd                     '// DAControlのSoundプロパティにセットする
  Call Window_OnResize()             '// コントロールの初期サイズ設定
  End Sub
'―――――――――――――――――――――――――――――――――――
'// 画面サイズ変更時の再描画
'―――――――――――――――――――――――――――――――――――
Sub Window_OnResize()
On Error Resume Next
  DAControl.Width  = Document.Body.ClientWidth
  DAControl.Height = Document.Body.ClientHeight
  '// クライアントの幅と高さのビヘイビアを変更(m.Pixel = 0.000263)
  ch.SwitchToNumber Document.Body.ClientHeight * m.Pixel.Extract / 2.1
  DAControl.focus
End Sub

'―――――――――――――――――――――――――――――――――――
'// Shapes DirectX 3Dトランスフォームで立方体のDAImageを作成する
'―――――――――――――――――――――――――――――――――――
Private Sub Meshmake()
Dim dxt            'As DXTMSFT3Lib.HeightField
Dim res            'As IDADXTransformResult
Dim trn            'As DATransform3
Dim matte 'As DAMatte          '// イメージクリッピング(切り抜く)用のDAMatte
Dim tmpimg'As DAImage          '// 画像分割イメージ用の配列
Dim size1 'As Long             '// 分割画像のサイズ(メートル = m.Pixel.Extract * ピクセル)
Dim fx, fy'As Long             '// 画像の表示位置をずらす値
Dim i, j  'As Long

  '// DirectX Transform オブジェクトの作成 (Shapes)
  Set dxt = CreateObject("DX3DTransform.Microsoft.Shapes")
  '// DirectX Transform の適用
  Set res = m.ApplyDXTransform(dxt, Null, Null)
  res.PutBvrAsProperty "Shape", m.DAString("Sheet")
  '// キューブ配列の最終番号
  last = (GRID ^ 2) - 1
  '// 動的配列の要素を確保
  ReDim block(last)
  ReDim xang(last)
  ReDim yang(last)
  ReDim rotx(last)
  ReDim roty(last)
  ReDim number(last, 5)
  '// 分割画像サイズ
  size1 = m.Pixel.Extract * SIZE
  '// 画像をキューブの数で分割し、6面それぞれに画像を貼り付ける
  For i = 0 To last
    For j = 0 To 5
      Set mesh(j) = res.OutputBvr
      '// 分割イメージの作成
      '// クリッピング領域の設定
      Set matte = m.FillMatte(m.Rect(size1, size1))
      '// 画像イメージの位置を移動してコントロールの中央部を切り抜く
      '// SIZE × GRID より大きな画像でないと、倍率が一定でなくなる
      fx = (size1 * (GRID - 1) / 2) - size1 * (i Mod GRID)
      fy = size1 * (i ¥ GRID) - (size1 * (GRID - 1) / 2)
      Set tmpimg = tximg(j).Transform(m.Translate2(fx, fy))
      '// イメージにクリッピングを適用
      Set tmpimg = tmpimg.Clip(matte)
      '// 物体にイメージを貼り付ける(マッピング)
      Set mesh(j) = mesh(j).Texture(tmpimg.MapToUnitSquare)
      '// 画像が逆さまになるので反転する
      Set mesh(j) = mesh(j).Transform(m.Rotate3Degrees(m.XVector3, 180))
      Set mesh(j) = mesh(j).Transform(m.Translate3(0, 0, 1))
      Select Case j
        Case 1, 2, 3
          Set mesh(j) = mesh(j).Transform(m.Rotate3Degrees(m.XVector3, 90 * j))
        Case 4, 5
          Set mesh(j) = mesh(j).Transform(m.Rotate3Degrees(m.YVector3, 90 + 180 * (j - 4)))
      End Select
      number(i, j) = j
    Next
    '// 一つの立体に合成する
    Set block(i) = m.UnionGeometryArray(mesh)
    '// 軸回転動作をビヘイビアとして配列に格納
    Set rotx(i) = m.ModifiableBehavior(m.DANumber(0))
    Set roty(i) = m.ModifiableBehavior(m.DANumber(0))
    '// X軸回転を定義する
    Set block(i) = block(i).Transform(m.Rotate3Anim(m.YVector3, roty(i)))
    '// Y軸回転を定義する
    Set block(i) = block(i).Transform(m.Rotate3Anim(m.XVector3, rotx(i)))
    '// キューブを平行移動
    Set trn = m.Translate3((i Mod GRID) * 2 - (GRID-0.99), GRID-0.99-(i ¥ GRID) * 2, 0)
    Set block(i) = block(i).Transform(trn)
```



★今，PIECEでゲームを作っています。自信作ができたらベーマガに投稿しますね。（青森県南津軽郡・ていこく23歳）……【**編**：PIECEの投稿第1位の座は今月号で取られちゃいましたね。第2位をぜひ狙ってください。**影**：いろんな言語の投稿お待ちしています！】

```
  Next
  '// 一つのグループに合成する
  Set group = m.UnionGeometryArray(block)
End Sub

'――――――――――――――――――――――――――――――――――――――――
'// マウスクリック時の処理
'――――――――――――――――――――――――――――――――――――――――
Private Sub DAControl_MouseDown(ByVal button, _
                              ByVal KeyFlags, ByVal X, ByVal Y)
Dim No              'As Long  '// 先頭のキューブ画像配列NO
Dim i, j  'As Long

  '// マウスポインタの座標をコントロール内での座標へ変換
  cx = X - DAControl.Style.posLeft - Document.Body.ClientWidth / 2 + 100 * GRID / 2
  cy = Y - DAControl.Style.posTop - Document.Body.ClientHeight / 2 + 100 * GRID / 2
    If cx > 0 And cx < 100 * GRID And cy > 0 And cy < 100 * GRID Then
      '// 座標からビューブ番号を計算
      i = cx ¥ 100 + (cy ¥ 100) * GRID
      '// キューブの回転と配列番号を置き換える
      Call Cube_Rot(button, i, 1)
      '// 先頭のキューブ画像配列NOを取得します
      No = number(0, 0)
      '// 先頭のキューブの向きが合っているかしチェックする
        If No =< 3 And number(i, 1) = (No + 1) Mod 4 Or _
             No => 4 And number(i, 1) = 1 Then
            '// 整列したかチェックする
            For i = 1 To last
              For j = 0 To 1
                If number(0, j) = number(i, j) Then
                    If i = last And j = 1 Then
                      Select Case No
                        Case 0
                          MsgBox "No." & No + 1 & " クリア  この調子でがんばってね!",, "キュ
ーブパズル"

                        Case 1, 3
                          MsgBox "No." & No + 1 & " クリア  すご～い!",, "キューブパズル"
                        Case Else
                          MsgBox "No." & No + 1 & " クリア  お疲れ様!",, "キューブパズル"
                      End Select
                    End If
                Else
                    Exit Sub
                End If
              Next
            Next
        End If
    End If
End Sub

'――――――――――――――――――――――――――――――――――――――――
'// キューブの回転と配列番号を置き換える処理
'――――――――――――――――――――――――――――――――――――――――
Private Function Cube_Rot(ByVal Jiku, ByVal No, ByVal Rot)
'// Jiku(回転軸):1 = Y軸、2 = X軸
'// No:キューブ番号、 Rot:回転回数
Dim buffer'As Long  '// 配列番号の一時保存用変数
Dim i, j, k         'As Long

  For k = 1 To Rot
    '// 指定の軸でキューブを回転する
    If Jiku = 1 Then
      '// Y軸回転(右へ回転する、X軸回転すると、Y軸も回転する)
      roty(No).SwitchTo m.Interpolate(yang(No), yang(No) + PI / 2, SPEED)
      yang(No) = yang(No) + PI / 2
        If yang(No) = 2 * PI Then yang(No) = 0
    Else
      '// X軸回転(下へ回転する、Y軸回転しても、X軸は回転しない)
      rotx(No).SwitchTo m.Interpolate(xang(No), xang(No) + PI / 2, SPEED)
      xang(No) = xang(No) + PI / 2
        If xang(No) = 2 * PI Then xang(No) = 0
    End If
    '// 回転軸と配列の番号から、主平面が、配列の何番目か調べて
    '// 順序格納ナンバーを検索し、配列番号を置き換える
    '// Y軸は、X軸回転により、軸自体も回転するが、X軸は、この様な事はない??
    If Jiku = 2 Then
      i = 4          '// マウス右ボタン(X軸回転)なら回転方向は全て同じ
    Else
      '// Y軸の主平面は、X軸と重なる事はない(0～3のいづれかにある)
      For i = 0 To 2
        If number(No, i) = Jiku + 2 Then Exit For
      Next
    End If
    '// 配列番号置き換え順序格納ナンバー順に配列番号の置き換え
    buffer = number(No, Mid(poschg(i), 1, 1))
    For j = 1 To 3
      number(No, Mid(poschg(i), j, 1)) = number(No, Mid(poschg(i), j + 1, 1))
    Next
    number(No, Mid(poschg(i), j, 1)) = buffer
  Next
End Function

'――――――――――――――――――――――――――――――――――――――――
'// キューブをランダム回転
'――――――――――――――――――――――――――――――――――――――――
Private Sub Rnd_Shuffle()
Dim val             'As Long  '// キューブ回転回数(90度)
Dim i, j  'As Long

  '// 乱数発生ルーチンを初期化します。
  Randomize
  For i = 0 To last
    '// X軸:1 とY軸:2 を1回ずつ、回転させる
    For j = 1 To 2
      val = Int(3 * Rnd)      '// 0 から 2 までの乱数を発生させます。
      '// 乱数の数だけキューブの回転と配列番号を置き換えをする
      Call Cube_Rot(j, i, val)
    Next
  Next
End Sub

'――――――――――――――――――――――――――――――――――――――――
```

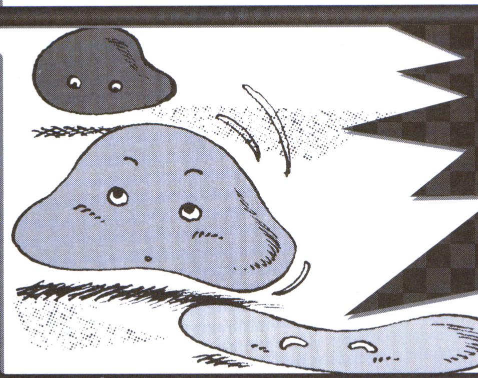
★「ダンシングベイビー」に対抗して「ダンシング影さん」発売決定か!?(千葉県山武郡・**野潤上15歳**)……【**つぐ美**:どうしたんですか,おしり振り振りして。**影**:これぞ「ダンシング影さん」。読者にお見せできないのが残念。**つぐ美**:誰も見たくありませんって。】

```
'// 先頭のキューブに合わせて整列する
'―――――――――――――――――――――――――――――――――――――
Private Sub Cube_Lineup()
Dim No          'As Long   '// 先頭のキューブ画像配列NO
Dim chkno 'As Boolean       '// 配列の中に、同じNOが有るかの判定
Dim i, j, k         'As Long

 '// 先頭のキューブ画像配列NOを取得します
 No = number(i, j)
 For i = 0 To last
  '// 先頭のキューブ画像配列NOと違うなら同じにする
  If number(i, 0) <> No Then
   chkno = False
   For j = 1 To 3
    If number(i, j) = No Then
      chkno = True
      Exit For
     End If
   Next
     '// 配列NOの１～３の中に、同じ番号がなければ、Y軸回転する
     If chkno = False Then Call Cube_Rot(1, i, 1)
     // 先頭のキューブ画像配列NOと違うなら同じにする
     If number(i, 0) <> No Then
     '// 先頭のキューブ配列NOと同じになるまでX軸回転する
     For j = 1 To 3
       Call Cube_Rot(2, i, 1)
       If number(i, 0) = No Then Exit For
     Next
     End If
  End If
  '// 向きを合わせる処理
  If No =< 3 And number(i, 1) <> (No + 1) Mod 4 Or _
    No => 4 And number(i, 1) <> 1 Then
   If No = 1 Or No = 3 Then
     '// Y軸主平面番号なら、Y軸回転をして、向きを整列
     For j = 1 To 3
      Call Cube_Rot(1, i, 1)
       If number(i, 1) = (No + 1) Mod 4 Then Exit For
     Next
   Else
    '// Y軸を回転する
     Call Cube_Rot(1, i, 1)
     '// X軸を２回転する
     Call Cube_Rot(2, i, 2)
     For k = 0 To 2
      '// 再度、Y軸を３回転し、元に戻して整列
      Call Cube_Rot(1, i, 3)
     Next
      '// 先頭のキューブ画像配列NOと違うならX軸を２回転する
      If number(i, 0) <> No Then Call Cube_Rot(2, i, 2)
   End If
  End If
 Next
End Sub

'―――――――――――――――――――――――――――――――――――――
'// メッセージ文字イメージの作成
'―――――――――――――――――――――――――――――――――――――
Private Sub Dispmesg()
Dim fnt          'As DAFontStyle          '// フォント
Dim tmpimg'As DAImage              '// メッセージの一時保存用変数

 Set ch = m.ModifiableNumber(0)
 '// イエロー・ＭＳ 明朝フォント・20ポイント・太字のDAFontStyleオブジェクトを作成
 Set fnt = m.DefaultFont.Color(m.Yellow).Family("ＭＳ 明朝").Size(20).Bold
 '// 文字列からDAImageオブジェクトを作成
 Set messimg = m.StringImage("マウス左ボタン：Y軸回転、マウス右ボタン：X軸回転", fnt)
 '// 常に画面下側で表示（ビヘイビアを使用）
 Set messimg = messimg.Transform(m.Translate2Anim(m.DANumber(0),m.Mul(ch, m.DANumber(-1))))
 Set tmpimg = m.StringImage("Space キーでシャッフル・降参ならESC キー", fnt)
 Set tmpimg = tmpimg.Transform(m.Translate2Anim(m.DANumber(0), m.Add(m.Mul(ch, m.DANumber(-
1)), m.DANumber(0.01))))
 '// メッセージ文字イメージの合成
 Set messimg = m.Overlay(tmpimg, messimg)
 Set tmpimg = m.StringImage("クリックして下さい。", fnt)
 Set tmpimg = tmpimg.Transform(m.Translate2Anim(m.DANumber(0),m.Sub(ch,m.DANumber(0.018))))
 Set messimg = m.Overlay(tmpimg, messimg)
 Set fnt = m.DefaultFont.Color(m.Red).Family("ＭＳ 明朝").Size(32).Bold
 Set tmpimg = m.StringImage("ＩＥ５ＤＡＣｏｎｔｒｏｌクラブ", fnt)
 Set tmpimg = tmpimg.Transform(m.Translate2Anim(m.DANumber(0),m.Sub(ch,m.DANumber(0.008))))
 Set messimg = m.Overlay(tmpimg, messimg)
End Sub

'―――――――――――――――――――――――――――――――――
'// DAControl_KeyDown : キー操作に対応する処理
'―――――――――――――――――――――――――――――――――
 Private Sub DAControl_KeyDown(ByVal KeyCode, ByVal KeyData)

  '// KeyUpイベントは、無視する
   If KeyData < 0 Then Exit Sub
  '// キーコードによる分岐
  Select Case KeyCode
   Case 32   '// [SpaceBar]
    '――――――――――――――――――
    '// キューブをランダム回転
    '――――――――――――――――――
     Call Rnd_Shuffle
   Case 27   '// [ESC]
    '――――――――――――――――――
    '// キューブを整列
    '――――――――――――――――――
     Call Cube_Lineup()
  End Select
 End Sub

</SCRIPT>
<! DirectAnimation コントロール>
 <OBJECT ID="DAControl"
  STYLE="position:absolute; top:0; left:0; z-index:-1"
  CLASSID="CLSID:B6FFC24C-7E13-11D0-9B47-00C04FC2F51D">
 </OBJECT>
</body>
</html>
```



★生まれた日が近づくと思い出すが…そろそろ年齢に1足さなきゃ。あれ？この変数，低は1だっけ，0だっけ？そもそもオレ27か？これ定数になってやしないか？（愛知県名古屋市・**わいち301？歳**）……【**つぐ美**：あら，私の年齢，ベーマガ創刊時から定数よ。】

# Excel(VBA)用



VBAでVB投稿作品を再現

移植版

# パニックスライム

内田 寛（53歳）



入力方法は49ページをご覧ください。

## はじめに

アイデアが乏しいので，今回も移植版です。本誌2000年11月号掲載の峠　恒司さん作『パニックスライム』は，ゲーム性もいいしキャラがかわいいので，これをExcelで出来ないかと考えました。

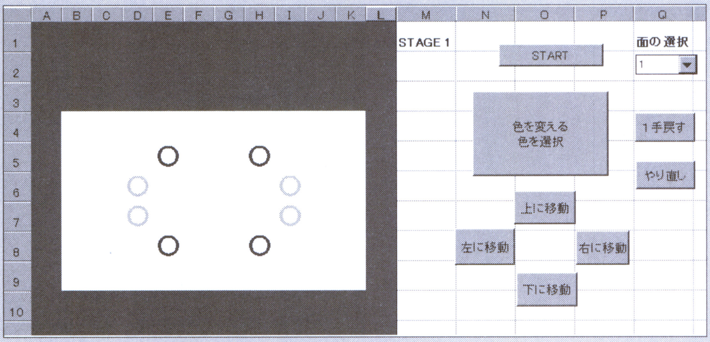
原作より，機能的には劣りますが，数式などを工夫してプログラムを短くすることができました。

## 遊び方・ルール

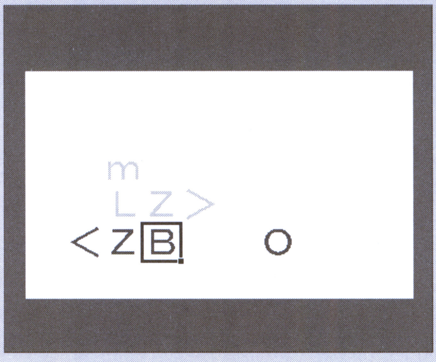
画面右側の「面の選択」ボックスで面の数を決めて「START」ボタンを押します。

画面左側に表示されているスライムを一つだけ選び（クリックする），「色を変える色を選択」ボタンを押します。「上下左右に移動」ボタンを押し,移動します。

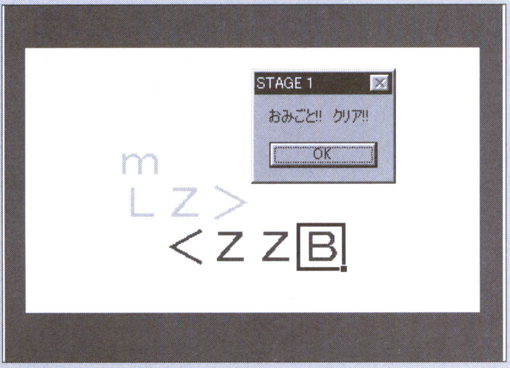
同じ色のスライムを合体させていきますが，途中で色を変えるときは他の色のスライムをクリックして色を変えるボタンを押します。今までのスライムは目を閉じ，新しいスライムが目を開き，移動



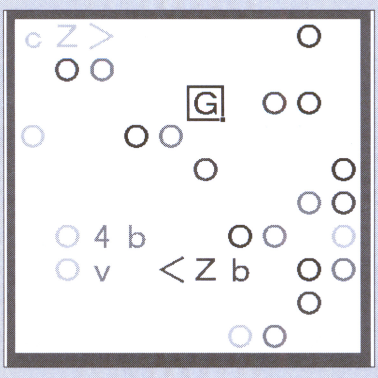

《写真１》画面右側のボタンを使用してゲームを遊ぶ



《写真２》ルールはVB版と同じで，違和感はない




《写真３》同じ色同士のスライムをくっつければＯＫ



《写真４》難易度の高いステージも用意されている

★パソコン・ユーザーにとってのゴールって，結局，何なのだろう？（茨城県古河市・**年齢不相応42歳**）……【**Dr.D**：ワシもパソコン暦は長いほうじゃが，ゴールは見えんのう。ゴールがなくたっていいんじゃないかのう。**影**：残り人生の少ないDr.からのお言葉でした。】

可能になります。すべてのスライムが合体するとクリアです。

「一手戻す」ボタン，「やり直し」ボタンもあります。1から30面が原作のレッスンコース，40，50，68面がチャレンジコースです。14面から23面までは，私が追加しました。

## セルの設定など

まずセルA1〜L12の範囲のフォントを28にして，行の高さ27.00，列幅4.00，書式設定を中央揃えにします。セルQ1に“面の数”，セルU1からU33までに数字の1〜30，40，50，68を記入した後，表示→ツールバー→フォームからボタンを8個貼り付けます。

マクロの登録ダイアログで新規作成，これをそのつど8回します。

コンボボックスを貼り付けて，右クリック→コントロールの書式設定ダイアログで入力範囲をU1:U33，リンクするセルをT20→OK。

ツール→マクロ→編集で，掲載されているリストを書いていきます。

## スライム表示設定

O・G・g…はじめ活動中の，ねむりの長さ1の各スライム

C・M・B・W…それぞれ左，上，右，下向きの活動中の頭

c・m・b・w…左，上，右，下向きのねむりの状態

Z・H…胴

2・L・」・4…まがりの状態

>・v・<・^…尾

を表します。

このままでもゲームはできますが，記述がめんどうなら，

O…◎

G・C・B・M・W…●

g・c・b・m・w…○

その他…■

にするのが簡単で見やすくゲームができます。

## CHECKER FLAG

**編**：セルをうまくマップとして利用していますね。テキスト表示なので，スライムの状態が少しわかりにくいのが難点ですが，VBAゲームとしてはよく出来ています。

**Dr.D**：リストは比較的短めではあるが，if文が大量に使用されている点が気になるのぉ。配列やループ文，もしくはselect文などをうまく利用すればもっとスッキリしたリストになるはずじゃ。ゲーム内容はしっかりと形になっているので，今後はリスト記述面にも気を配ってみてはどうかな？

**つぐ美**：ダウンロードファイルには，スライムの表示形態を変更したバージョンが三つ収録されています。それぞれ，デフォルトのアルファベット表示，記号を使用した簡略化バージョン，そして誌面には掲載しませんでしたが，外字を使用したバージョンが収録されています。あなたの目で違いを確認してみてくださいね。ちなみにわたしのオススメは，記号を使用しているバージョンね。スライムがスッキリと表示されるので，遊びやすいわよ。

## コラム　若きクリエイターの熱気！

### iアプリゲームコンテスト開催される!!

NTTドコモ東北は3月23日，仙台市の仙台サンプラザで「第1回ｉアプリゲームコンテスト　ドコモカップ東北」の最終審査会を開催しました。グランプリに選ばれたのはカラーパズルゲーム「PALETTE」(チーム＝COLORS，国立仙台電波工業高等専門学校)。当日は表彰式のほか，講演会，各社展示，ゲストによるステージトークなども行われ，約550人が来場しました。

同コンテストは東北6県の高校，大学，大学院，専門学校の学生を対象に次世代クリエイターの創出などをねらい，同社モバイルマルチメディアビジネス部などが企画したもので，全国でも初の試み。

昨夏時点で95作品110人の応募があり，書類選考で24組43人，2次審査を経て今回の6作品15人に絞り込まれました。最終審査は，①ドコモ公式サイト上からのダウンロード·投票，②当日会場で体験·投票，③審査員(13人)の判定，の三つの総合ポイントで競われました。公式サイトには2か月で約15万4千のアクセス，6万5千件のダウンロードがありました。

審査員はｉアプリの特性と高いエンターテイメント性，独創性，将来の可能性などを基準に選考，グランプリのほか準グランプリ2作品「ブロッカー」(福島県立会津大学)，「バネクライシス」(仙台電波高専)と，優秀賞3作品「どすこい相撲部屋」(八戸工業高専)，「天下人」(山形大学大学院)，「スティックスタック」(岩手県立大学)と確定，受賞作には副賞のほか，ドコモカップ認定のアイコンが進呈され，入賞者のサイトに張ることもできます。

ステージでは，講演「ｉアプリ制作の課題と今後」(木寺祥友エル・カミノ・リアル社長＝審査員)，ファイナルデモステージ，抽選会(進行役＝坂本ちゃん，真鍋かをり)なども行われました。また，展示·体験フロアでは端末メーカー5社(ソニーエリクソン，ＮＥＣ，富士通，松下，三菱)と，ＩＰ4社(コナミ，サイバード，ナコム，ハドソン)がブースを設け，ｉアプリ対応10機種を集めたコーナーで，同コンテストゲームの体験·投票も行われました。

グランプリの「PALETTE」のタイトル画面

若いクリエイターが集まった最終審査会場

★うを，もう1か月過ぎたのか，今日も本屋へGO！GO！（愛知県一宮市・**幻19歳**）……【**つぐ美**：ベーマガと出会うと，楽しさのあまり月日のたつのも早いわよね。**影**：もう5月号だもんね。2003年ももうすぐだ！つぐ美ちゃんも年をとるわけだ。**つぐ美**：ばきっ！】

## 《リスト》パニックスライムのプログラム・リスト

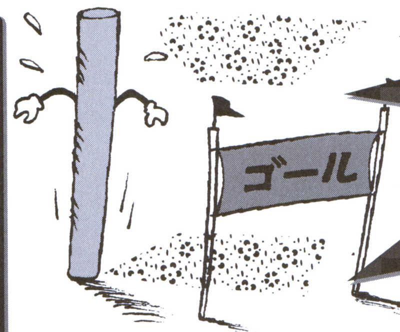
```
Dim G, W, E, M, M1, N, N1, M2, N2, K, F, Q, R, S, T, C, C2, Z$(30), I(4), D(30, 30)
Dim A(30, 30), B(30, 30), X(40, 4), Y(40, 4), X2(40, 4), Y2(40, 5), L(4), L2(4)
Sub ボタン1_Click()
  I(0) = 3: I(1) = 5: I(2) = 4: I(3) = 1: I(4) = 6: E = 0  'I()各スライムの色 :Eは空白と枠
  Randomize: G = Cells(Cells(20, 20), 21): U$ = "7777777777": U2$ = U$ & U$: U3$ = U2$ & U$
   V$ = "0001000002111110202000110002000101002020100102000 0"
Z$(0) = V$ & "033304000000000040000333344055003004040003022400 00"
Z$(1) = U2$ & "0000000000000200200000300003000030000300000200200000000000000" & U2$
Z$(2) = U2$ & "70201000777077770777010001007777177077771002010 17" & U3$
Z$(3) = U3$ & "7111772227711177222770000000077222771107" & U3$
Z$(4) = U3$ & "7771210777777000277777730137777771302777" & U3$
   V$ = "1207777777300777777777000777777700077777777200777"
Z$(5) = V$ & "7777310777777777000777777700077777777701777777723"
Z$(6) = U$ & "71004002777077077077707403707773000001777071027077707707707774002003 77" & U2$
Z$(7) = U$ & "77107710177701200207777110101777011201077712001177777700207777711 0177" & U2$
Z$(8) = U3$ & "1207777012210000001277077770777707777077770000077" & U2$
   V$ = "7777277777777400777777201027777301000377102034043 77"
Z$(9) = V$ & "1020142777731043777777000777777774777 77" & U$
   V$ = "1230703217230070032730007000370000000000777700077 77"
Z$(10) = V$ & "0000000007300070003723007003271230703217" & U$
Z$(11) = U$ & "72110777777111077777711107777770000000077777702227777770222777777022 17" &
U2$
Z$(12) = U3$ & "21100002221110770222111077022211100002 21" & U3$
    V$ = "7777017777777702777777770377777777047771234000000 00"
Z$(13) = V$ & "00004321777740777777773077777777207777777710 7777"
    V$ = "1005004002077077077007707707703001002004077077077 0"
Z$(14) = V$ & "07707707704002001003077077077007707707702004005001"
Z$(15) = U$ & "101055010217077770711707777071103333320127077770732707777073204045 0403" &
U2$
Z$(16) = U2$ & "77311111377771777177777177717777710771777711177177771021117702107717 77" &
U$
Z$(17) = U$ & "227007744204717370104000000202347707077733770702477720074007442047021 7" &
U2$
    V$ = "11111144442222770774333377077430050022244770770774"
Z$(18) = V$ & "47707707744442005003277077333327707722222224111111"
    V$ = "5030440302070700707043337705330733775570077700777 0"
Z$(19) = V$ & "40070070040704005070302277140107074470702030000302"
Z$(20) = U$ & "57217517470000200222073117777224777322220700107272270070722200010133 22" &
U2$
Z$(21) = U$ & "33371147772037014777303007700323070500237777707220714004722574014477 77" &
U2$
    V$ = "12340043212234774322333477433344440044440770770770"
Z$(22) = V$ & "07707707704444004444333477433322347743221234004321"
    V$ = "50050050050771771770333333333333771771773333333333"
Z$(23) = V$ & "27717717702004224002177077077017707707705055555505"
    V$ = "77771500007777027770777037770777704777012340000055"
Z$(24) = V$ & "55000004321077740777707773077770777207777000015 7777"
    V$ = "03047777771020777777030477777750107777770200054321 3"
Z$(25) = V$ & "04077777705017777772030777777040577777710207777 77"
Z$(26) = U3$ & "2110000221311077022331107702232110000221" & U3$
Z$(27) = U2$ & "1111005555333300222222220011114444004444555500333 3" & U3$
    V$ = "12047402172040704027040070040740000000477770007777 4"
Z$(28) = V$ & "0000000470400700407204070402712047402177777777777"
Z$(29) = U$ & "71004002777577077577707403707773000001777071027077707707707774002035 77" &
U2$
    V$ = "15507705545550770555555007055500000000000777012077 77"
Z$(30) = V$ & "77034077700000000000555070055555507705552550770553"
  For M = 0 To 30: For N = 0 To 30: A(M, N) = 200: Next: Next: MAKE: REPL '画面をかく
End Sub
Sub ボタン2_Click()
M1 = ActiveCell.Column - 1: N1 = ActiveCell.Row - 1: H = A(M1, N1)
Select Case W
 Case 0  'W=0は１番最初のクリック：Hはクリックした位置のDATA：PUTAへはM,N,Kの値をわたす。
  If H = -1 Or H = 200 Then Exit Sub Else W = 1: MSAV: M = M1: N = N1: K = H + 5: PUTA:
SET1
 Case 1 'H=10から14は長さ１のねむり35から104は長さ２以上のねむりをクリック
  If H > 9 And H < 15 Then CNG: M = M1: N = N1: K = H - 5: PUTA: M2 = M: N2 = N: Exit Sub
  If H > 34 And H < 105 Then
   If H Mod 5 = C Then Exit Sub '同じ色:無視、違うとき:活動中はねむり、選択されたのが起き
る
    Call CNG: C = H Mod 5: M = X(1, C): N = Y(1, C): K = A(M, N) - 70: PUTA: M2 = M: N2 =
N
  End If 'H:0から4はすでに活動中のスライムがあり、それとは色が違う長さ１のスライムをクリッ
ク
  If H > -1 And H < 5 And L(H Mod 5) = 0 Then MSAV: CNG: M = M1: N = N1: K = H + 5: PUTA:
SET1
End Select
End Sub
Sub PUTA()
  If K = 200 Then E = 10: IRO: E = 0: Cells(N + 1, M + 1) = "" '枠
  If K = -1 Then IRO: Cells(N + 1, M + 1) = "" 'K=-1:空白、以降Kの値でスライムの色と形を分
類
   If K > -1 And K < 5 Then C = K Mod 5: IRO: Cells(N + 1, M + 1) = "O"
   If K > 4 And K < 10 Then C = K Mod 5: IRO: Cells(N + 1, M + 1) = "G"
   If K > 9 And K < 15 Then C = K Mod 5: IRO: Cells(N + 1, M + 1) = "g"
   If K > 14 And K < 20 Then C = K Mod 5: IRO: Cells(N + 1, M + 1) = "C" '起きた頭
   If K > 19 And K < 25 Then C = K Mod 5: IRO: Cells(N + 1, M + 1) = "M"
   If K > 24 And K < 30 Then C = K Mod 5: IRO: Cells(N + 1, M + 1) = "B"
   If K > 29 And K < 35 Then C = K Mod 5: IRO: Cells(N + 1, M + 1) = "W"
   If K > 34 And K < 40 Then C = K Mod 5: IRO: Cells(N + 1, M + 1) = "Z" '胴
   If K > 39 And K < 45 Then C = K Mod 5: IRO: Cells(N + 1, M + 1) = "H"
   If K > 44 And K < 50 Then C = K Mod 5: IRO: Cells(N + 1, M + 1) = "2" '曲がり
   If K > 49 And K < 55 Then C = K Mod 5: IRO: Cells(N + 1, M + 1) = "L"
   If K > 54 And K < 60 Then C = K Mod 5: IRO: Cells(N + 1, M + 1) = "」"
   If K > 59 And K < 65 Then C = K Mod 5: IRO: Cells(N + 1, M + 1) = "4"
   If K > 64 And K < 70 Then C = K Mod 5: IRO: Cells(N + 1, M + 1) = ">" '尾
   If K > 69 And K < 75 Then C = K Mod 5: IRO: Cells(N + 1, M + 1) = "v"
   If K > 74 And K < 80 Then C = K Mod 5: IRO: Cells(N + 1, M + 1) = "<"
   If K > 79 And K < 85 Then C = K Mod 5: IRO: Cells(N + 1, M + 1) = "^"
   If K > 84 And K < 90 Then C = K Mod 5: IRO: Cells(N + 1, M + 1) = "c" '寝た頭
   If K > 89 And K < 95 Then C = K Mod 5: IRO: Cells(N + 1, M + 1) = "m"
   If K > 94 And K < 100 Then C = K Mod 5: IRO: Cells(N + 1, M + 1) = "b"
   If K > 99 And K < 105 Then C = K Mod 5: IRO: Cells(N + 1, M + 1) = "w"
   A(M, N) = K 'セルにかいた後、仮想MAPのDATAもかきかえる
```

★僕のWebサイトがついにできました。アドレスは，http://isweb37.infoseek.co.jp/area/kokihiro/webworldです。でも，掲示板の書き込みが1件もありません（；；)。（福島県郡山市・**近内沖尋19歳**）……**【つぐ美：HP見ました。私おこってませんよ。】**

```
End Sub
Sub ボタン3_Click()
  For N = 1 To 10: For M = 1 To 10: K = D(M, N): PUTA: Next: Next 'セルにかく
  For N = 0 To 4: L(N) = L2(N): For M = 1 To L(N) '1手前のDATAにかきかえ
    X(M, N) = X2(M, N): Y(M, N) = Y2(M, N): Next: Next 'X(先頭から何番目,色)=X座標
  C = C2: M2 = X(1, C): N2 = Y(1, C): Cells(N2 + 1, M2 + 1).Select 'セルを頭にもっていく
End Sub
Sub ボタン4_Click()
  M1 = M2 - 1: N1 = N2: MSAV: BUNKI: KAKUNIN 'M2,N2:活動中のスライムの座標
End Sub
Sub ボタン5_Click()
  M1 = M2 + 1: N1 = N2: MSAV: BUNKI: KAKUNIN 'M1,N1これから進む先
End Sub
Sub ボタン6_Click()
  M1 = M2: N1 = N2 - 1: MSAV:  BUNKI: KAKUNIN
End Sub
Sub ボタン7_Click()
  M1 = M2: N1 = N2 + 1: MSAV: BUNKI: KAKUNIN
End Sub
Sub CNG()
  M = X(1, C): N = Y(1, C): J = A(M, N) 'ねむりをかく、長さ1と2以上で分岐
  If J > 4 And J < 10 Then K = J + 5: PUTA Else K = J + 70: PUTA
End Sub
Sub IRO()
  Cells(N + 1, M + 1).Select: Selection.Font.ColorIndex = I(C) 'FONTの色など
  Selection.Interior.ColorIndex = E: Selection.Interior.Pattern = xlSolid
End Sub
Sub LN1()
  M = M1: N = N1: H = M1 - M2: J = N1 - N2 '長さ2のとき頭を決め、尾を決めてかく
  If Abs(H) = 1 Then K = 5 * H + 20 + C: PUTA: M = M2: N = N2: K = S * H + Q + C: PUTA
  If Abs(J) = 1 Then K = 5 * J + 25 + C: PUTA: M = M2: N = N2: K = T * J + R + C: PUTA
End Sub
Sub LN2()
  H = A(M2, N2) '長さ3以上のとき頭と2番目を決める。頭の向きで分類
  If H > 14 And H < 20 Then Q = 37.5: R = 55: S = 2.5: T = 5:  LN1
  If H > 19 And H < 25 Then Q = 52.5: R = 45: S = 7.5: T = 5:  LN1
  If H > 24 And H < 30 Then Q = 30: R = 50: S = 5: T = -5: LN1
  If H > 29 And H < 35 Then Q = 52.5: R = 35: S = -2.5: T = 5: LN1
End Sub
Sub SET2()
  X(1, C) = M1: Y(1, C) = N1: X(2, C) = M2: Y(2, C) = N2: L(C) = 2
  M2 = M1: N2 = N1: Cells(N2 + 1, M2 + 1).Select '長さ2のとき頭と尾を記録
End Sub
Sub SET3()
  For J = L(C) To 2 Step -1: X(J, C) = X(J - 1, C): Y(J, C) = Y(J - 1, C): Next
  X(1, C) = M1: Y(1, C) = N1: M2 = M1: N2 = N1: Cells(N2 + 1, M2 + 1).Select
End Sub '長さ3以上のときの記録
Sub LN3()
  M = X(L(C) - 1, C): N = Y(L(C) - 1, C): M2 = X(L(C) - 2, C): N2 = Y(L(C) - 2, C)
  If Abs(M2 - M) = 1 Then K = 70 + 5 * (M2 - M) Else K = 75 + 5 * (N2 - N)
  K = K + C:  PUTA '尾を決めてかく
  M = X(L(C), C): N = Y(L(C), C): K = -1: PUTA '今までの尾の位置に空白をかく
End Sub
Sub MAKE()
 If G <> 0 Then Cells(1, 13) = "STAGE" + Str$(G) Else Cells(1, 13) = "おめでとう"
 If G < 31 Then '画面DATAよりB(M,N)を決める
   H = 0: For N = 1 To 10: For M = 1 To 10
   H = H + 1: K = Val(Mid$(Z$(G), H, 1)): If K = 7 Then K = 201 '画面DATAを分解
   B(M, N) = K - 1: Next: Next 'PUTAで K MOD 5を使うため K-1とした
 Else
   For M = 1 To 10: For N = 1 To 10: B(M, N) = -1: Next: Next '40,50,68面はランダムで
   For J = 1 To G  'G= 40  or  50  or  68
     Do: M = Int(Rnd * 10) + 1: N = Int(Rnd * 10) + 1: Loop While B(M, N) > -1
   K = Int(Rnd * 5): B(M, N) = K: Next 'ここのKはスライムの色
 End If
End Sub
Sub BUNKI()
 If A(M1, N1) = -1 Then '=-1:進む先が空白のとき
   Select Case L(C) 'その色のスライムの長さで分岐
   Case 1: K = C + 5: M = M1: N = N1: H = M2: J = N2: PUTA: SET1 '長さ1:空白の位置に移動
       K = -1: M = H: N = J: PUTA: Cells(N2 + 1, M2 + 1).Select '今までいたところは空白
   Case 2: Q = 70: R = 75: S = 5: T = 5: LN1: M = X(2, C): N = Y(2, C): K = -1: PUTA: SET2
   Case Else: LN2: LN3: SET3 '長さ3以上のスライム:頭と2番目,尾,今までの尾の位置に空白かく
   End Select: Exit Sub
 End If
 If A(M1, N1) = C Then  '=C:同じ色の長さ1のスライム
   If L(C) = 1 Then Q = 70: R = 75: S = 5: T = 5: LN1: SET2 Else LN2: L(C) = L(C) + 1: SET3
 End If  '長さ1のときは頭と尾の長さ2になる。2以上のときは頭と2番目が変化、尾は不動
End Sub
Sub MSAV()
 C2 = C: For M = 1 To 10: For N = 1 To 10: D(M, N) = A(M, N): Next: Next '動く前に1手記録
 For N = 0 To 4: L2(N) = L(N): For M = 1 To L2(N)  'C2活動中の色、L2()各色の長さ記録用
 X2(M, N) = X(M, N): Y2(M, N) = Y(M, N): Next: Next 'X2(,),Y2(,)は各色全ての座標記録用
End Sub
Sub KAKUNIN()
 F = 0: For M = 1 To 10: For N = 1 To 10: If A(M, N) > -1 And A(M, N) < 5 Then F = 1
 Next: Next 'まだ長さ1のスライムがあるとF=1
 If F = 0 Then
   MsgBox "おみごと!!  クリア!!", 0, "STAGE " + CStr(G)
   If G < 30 Then
     G = G + 1: Call MAKE: REPL: Exit Sub 'Gは画面数、G+1で次の画面へ
   ElseIf G = 30 Or G = 68 Then MsgBox " 全面クリア!!": G = 0: MAKE: REPL: Exit Sub
   End If
 End If
End Sub
Sub SET1()
 X(1, C) = M: Y(1, C) = N: L(C) = 1: M2 = M: N2 = N '長さ1の活動中のスライム記録
End Sub
Sub REPL()
 For M = 1 To 10: For N = 1 To 10: A(M, N) = B(M, N): Next: Next 'B(,)は始めのDATA
 For J = 0 To 4: L(J) = 0: Next: M2 = 1: N2 = 1 '各色のスライムの長さL()を0にもどす
 For N = 0 To 11: For M = 0 To 11: K = A(M, N): PUTA: Next: Next: W = 0 '画面をかく
End Sub
Sub ボタン8_Click()
 REPL 'GIVE  UPのとき
End Sub
```

★バーチャルベマガアイドル・つぐ美20歳。(和歌山県橋本市・ヅラChu19歳)……【つぐ美:バーチャルとは,日本語で「仮想」なんていうふうに訳す場合が多いけど,辞書を引くと「事実上の」とか「実際の」という意味もあるのよ。勉強になったでしょ。】

## Windows(F-BASIC V6.0)用




チョット懐かしのあのゲーム

# 感撃(かんげき)！イロイロ棒(ぼう)


MONOMON（32歳）




### ごあいさつ

この世にイライラ棒ゲームは数あれど，マップの仕掛けに凝ってはいても,棒にスポットを当てたものはあったでしょうか？

今回，主役のはずなのに日陰を歩んできた棒たちに自己主張してもらいました。

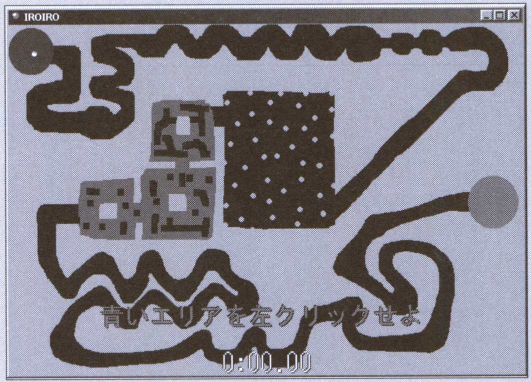
なお，「感撃」とは「感激」をもじった言葉で，「感電」と「電撃」をくっつけたものです。失敗時に少しでもビックリしてもらえたらと凝ってみた電気効果にちなんで名づけました。

### 遊び方

タイトル画面で左クリックすると棒セレクト画面になります(**写真1**)。

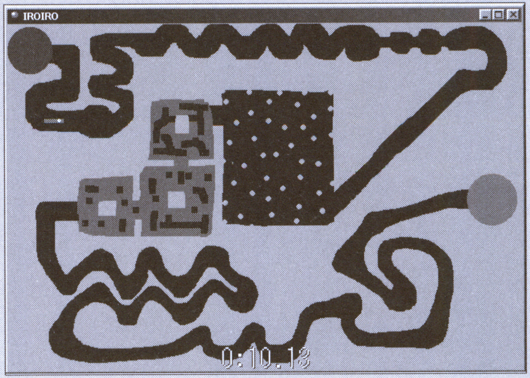

入力方法は47ページをご覧ください。

《写真1》今ではチョット懐かしい感じのあるゲーム



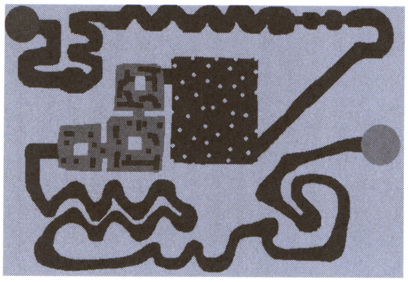

《写真2》壁にぶつからないように棒をマウスで操作

《写真3》まずは基本的な棒でのクリアを目指そう

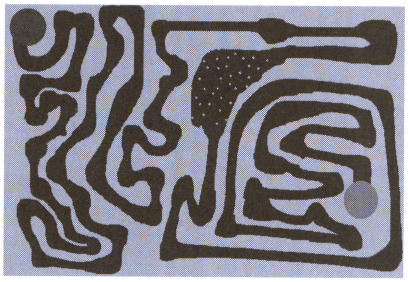



《写真4》数多くの種類がある棒がこのゲームのウリ

★言語制作の連載は，少し前にやったので，ちがったアプローチを知ることができて，おもしろいです。（福島県伊達郡・**菅野友紀18歳**）……【**編**：うれしいお便りですね。連載もいろんなテーマでやっていきます。みなさんの意見・感想，どしどしお寄せください。】



IROIRO.bmpの画像



IROIRO2.bmpの画像

雷をマウスで動かし，使いたい棒に合わせると選択状態になり，左右のボタンを同時に押すと決定します。選択状態時は，試しに動かせる状態になっていて，マウスに合わせて動きを確認できます。また，左右のクリック時の動作も確認できます(同時押ししたら決定されるので注意)。

次にマップが出てくるので，棒を青いエリア(スタート地点)に重ねて左クリックするとスタートします(**写真2**)。ゴールは赤いエリアです。壁に当たらないように棒を操作してゴールしてください(**写真3**)。

## デバッグ中の注意

ゲーム中にマウスを消去して移動範囲を制限する処理が入っています。このときにエラーが出て終了すると，マウスが消えたままになります。

そんな時は慌てず［Ctrl］+［Alt］+［Delete］を1回だけ押してください。「プログラムの強制終了」ウィンドウが出てくると同時にマウスが復活するので，そのウィンドウを「キャンセル」で閉じれば元どおりです（終了やシャットダウンを選択しないように注意！）。

## マップファイルについて

マップはビットマップ形式で640×432サイズで描いてください。

青はスタート，赤はゴールとなり，緑を含む色で輝度が&HC0以上の色(白，明るい灰，緑，水色，黄色)が壁となります。それ以外(黒，暗い灰，紫，暗いほうの各色)は通路となります。特に暗い灰色はミスにはならないけど，自分が見えにくくなる特殊な通路になります。

そうして作成したビットマップファイルを“IROIRO.bmp”のファイル名で，実行ファイルと同じフォルダに保存してください。

クリアできない面(赤いエリアがない)や，スタートできない面(スタートと同時に死なないように，棒が壁に重なった状態で青いエリアをクリックしてもスタートできないようになっているので，青いエリアが小さすぎた場合，スタートできない面になることがあります）でゲームを始めてしまった場合，［ESC］キーを押せばタイトルに戻ります。

マップファイルは，一応二つ作ってみました。“IROIRO2.bmp”は最初に作成した面で，部分的に面白い山場が多いのですが，長すぎて最後まで遊べないので“IROIRO.bmp”を作成しました。“IROIRO2.bmp”のマップで遊びたい場合は，ファイル名を“IROIRO.bmp”に変更してから，ゲームを実行してください。

## サウンドファイルについて

Program Filesフォルダのデータファイルを直接呼び出しています。どのファイルを呼び出しているかは，初期起動時にIROIRO.txtというファイルを作成し，そこに記述されています。

音を変更したい場合，このファイルの""で囲まれたファイル名を書き換えれば変更できます。

## 苦労したところ

はじめに考えたアイデアは風船，プロペラ，リボン棒の三つしかありませんでした。とりあえず作っていくうちに，いくつかの案が浮かんだのですが，頭の中で考える限り面白くなさそうだったので，それらは次々と没にしていました。

その時事件が起こりました。GBAのくるくるくるりんの発売です。いくらゲーム情報に疎くなったとはいえ，あれだけTVコマーシャルが流れれば目に入りま

### CHECKER FLAG

**つぐ美**：マウスで楽しめるゲームの典型的な例ね。基本内容は単純ですが，棒の種類の多さや，動きの個性などオリジナリティがしっかりしています。

**影**：ノートパソコンなどのパッドで遊ぶと，また違った感覚で楽しめるな。

**Dr.D**：注目してほしいのは，やはり特徴のある数々の棒の動きじゃな。リスト面では，各棒ごとにcase文で場合分けして処理しておるので，興味深い動きをしている棒の処理方法を，各自で研究してみると，いい勉強になるじゃろう。

**編**：三角関数をはじめ，いろいろな処理をしていますからね。内容を理解したら，オリジナルな動きをする棒を作っても面白いかもしれません。

★FM-7…まだウチにあります，っていうか動きます（笑）。（長野県上田市・**ぱんだまん@もあ20歳**）……【**Dr.D**：なつかしいのう。動くのはいいが，使ってるのかな？**影**：使ってなかったら頂戴って言いたいようですね，Dr.。だめですよ，読者から物せしめちゃ。】

す。あちゃー，かぶった！…風船とリボンしかオリジナル要素がなくなったのです。

それから考えつく限りのすべてを作ってみて，面白かったら採用する方式に変わりました。当然，没にしたアイデアも復活し…半分意地になりながら作りに作ってその数30以上！で，その中から12個を選出したのですが，リスト容量も選考の対象になったので，ぶっ飛んだやつは消えてしまいました(残念)。

没作品も遊んでみたいという奇特な方がおられましたら，完全バージョン(実験バージョンとも言う)もダウンロードファイルの方へ添付しましたので，遊んでみてください(全然ゲームになってないのもありますが，それは実験作品ということでご勘弁を…)。

## 改造など

できれば，棒を増やすのに挑戦してみてください。

作者は一応しぼりきったので満足しましたが，アイデアは無限にあるはずなのでまだまだ(プログラミングで)遊べると思います。

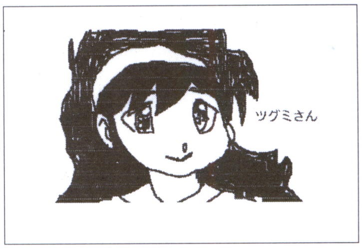
なお，棒の状態を記録できるフラグが二つの2倍の精度実数しかないので，その制限の中で凝ったやつを作ると気が狂いそうになります。ご注意ください(フラグが二つしかないのは，元々そんな凝った棒を作る気がなく，実験で作っても投稿バージョンで削除する予定だったからです)。

**★参考にしたもの**

・本誌1997年11月号『崩せ』　坂井弘亮作

・「Windows98 APIリファレンス for Visual Basic Programmers」(株）秀和システム　河北潤二：著

評…マウスで描いたそうです(つぐ美)




by 前田宣弘

《第1表》変数表

| | |
|---|---|
| 定数 | |
| PI=3.14159 | 円周率 |
| P_ALL | 棒の総数-1 |
| ※PAGEMAX | 棒の総ページ数-1 |
| ※PMAX(PAGEMAX) | そのページ内の棒の総数-1 |
| ※PST(PAGEMAX) | そのページ内の最初の棒の通しナンバー |
| 文字データ | |
| MES$(P_ALL+1,2) | 棒メッセージ<br>引数1[0:セレクトメッセージ,1以上:棒の番号+1],<br>引数2[0:識別子=棒の名前,1:見出し,2:解説] |
| COM$(3) | ゲーム中のコメント<br>0:TIMEUP,1:GAMEOVER,2:CLEAR,3:ハイスコア |
| SND$(4) | サウンドファイル名 |
| プレイヤー(棒)まわり | |
| F(P_ALL), G(P_ALL) | 棒ごとのフラグ<br>棒セレクト時に，同時に複数キープするため別々に用意 |
| PX(8),PY(8) | リボン棒専用配列(各リボンパーツの座標が入る) |
| ゲームメイン | |
| GF | ゲームフラグ<br>0:タイトル,1:棒セレクト,2:スタート前,3:ゲーム中,<br>4:タイムアップ,5:ゲームオーバー,6:クリアー,7:ハイスコア |
| P | 選択された棒ナンバー(棒セレクト時，-1で選択されていない) |
| T | ゲームタイマー |
| HI(P_ALL) | ハイスコア |
| KB$ | キー入力変数 |
| MX,MY,ML,MR | マウス座標(X,Y)，マウスボタン情報(左,右) |
| ※STF | ステージイメージ拡大フラグ<br>0:実サイズ,1:縦を揃える,2:横を揃える,3:縦横を揃える |
| ※PF | 現在の棒ページナンバー |
| ※ZM | ズームフラグ（0:拡大なし,1-6:倍率-1） |
| タイトル、棒セレクト時 | |
| X,Y | タイトル時の雷の座標 |
| X(P_ALL),Y(P_ALL) | 棒セレクト時の枠の表示座標 |
| 各種オブジェクト変数 | |
| BUF | ウラ画面用ビットマップオブジェクト |
| STG | ステージ用ビットマップオブジェクト |
| PL(1) | 棒用ビットマップオブジェクト(PL(1)はアタリ判定用バッファ) |
| MUS | マウスカーソル消去用空ポインタオブジェクト |
| FORM | フォームオブジェクト |
| その他 | |
| WINSIZ | マウス移動範囲設定用RECT構造体 |
| GTC_TIMER# | GetTickCount時間保存用変数 |
| DT | 単位時間(プログラム1ループの時間をミリ秒単位で取得) |
| F$,S$ | ステージファイル名，セーブファイル名 |
| A$,B$,A,B,C,D,I,J | 雑用 |

※マークは完全版にのみ使用されているものです

《リスト》感撃！イロイロ棒のプログラム・リスト

```
#include "WINDOWS.BI"
if PREVINSTANCE<>0 then end        '多重起動防止

type RECT
  X1 as long
  Y1 as long
  X2 as long
  Y2 as long
end type

'Windows API 関数宣言
declare function SPS& lib"winmm.dll" alias"sndPlaySoundA"(byval N$,byval F&)
declare function GTC& lib"kernel32" alias"GetTickCount"()
```

★かつての月刊マイコンでもあった，プログラミングテーマを決めて投稿を応募するというのをやってもらえないでしょうか？今だったらメールで簡単に応募できると思います。(島根県松江市・NSR250R＋横浜ナンバー38歳）……【編：いいですね，そのアイデア。】

```
declare function GSM& lib"user32" alias"GetSystemMetrics"(byval H&)
declare function CC1& lib"user32" alias"ClipCursor"(A as RECT)
declare function CC2& lib"user32" alias"ClipCursor"(byval A&)

'プロシージャ宣言
declare sub HI_SAVE()
declare sub LIGHT(S,X,Y,W,WD,F)
declare sub INA(byval X,byval Y,byval DV,byval W,byval WD,byval F)
declare sub LOGO(A$,Y,W,H,C)
declare sub MOUSECHECK(MX,MY,LC,RC)
declare sub MMCLIP(F)
declare sub MCWAIT()
declare function TIMER(WT)
declare function GAMETIME$(T)
declare function COL(C,L)
declare function HITCHECK(X,Y)
declare sub PLAYER(N,X,Y,F as double,G as double)


'**************************** ゲーム初期設定 ****************************
'ｸﾞﾛｰﾊﾞﾙ変数宣言、配列宣言
var shared BUF as object, STG as object, MUS as object, FORM as object
var shared F$, S$, PI, GTC_TIMER#, KB$, P_ALL
dim PL(1) as object, PX(8), PY(8), SND$(4), COM$(3)

'定数、ｵﾌﾞｼﾞｪｸﾄ設定
PI=3.14159
COM$(0)="ＴＩＭＥ ＵＰ": COM$(1)="ＧＡＭＥ ＯＶＥＲ"
COM$(2)="ＣＬＥＡＲ"  : COM$(3)="ハイスコア！"
for I=0 to 1: A=64-32*I: BITMAPOBJECT PL(I): PL(I).CREATEBITMAP A,A: next
BITMAPOBJECT BUF
BITMAPOBJECT STG: STG.CREATEBITMAP 640,432
POINTEROBJECT MUS
FORM.ATTACH GETHWND                    'ﾌｫｰﾑｵﾌﾞｼﾞｪｸﾄ

'棒のﾃﾞｰﾀ読込み
restore *PL: read P_ALL
dim X(P_ALL), Y(P_ALL), MES$(P_ALL+1,2), HI(P_ALL)
dim F(P_ALL) as double, G(P_ALL) as double
for I=0 to P_ALL+1: read MES$(I,0),MES$(I,1),MES$(I,2)
  if I>0 then MES$(I,1)=MES$(I,1)+"！"+MES$(I,0)+"棒"
next

'ｾﾚｸﾄ画面用座標計算
for I=0 to P_ALL
  A=(I/(P_ALL+1)-.25)*2*PI
  X(I)=320+280*cos(A): Y(I)=216+180*sin(A)
next

'ｽﾃｰｼﾞ読込み
F$="IROIRO.BMP"
A=0: on error goto *ERR: BUF.LOADBITMAPFILE F$: on error goto 0
if A then print "BMPﾌｧｲﾙでないか、壊れているか、存在しません": print F$: stop: end
STG.line(0,0)-(639,431),,14,bf,14
STG.DRAWBITMAP BUF,320-BUF.GETBITMAPWIDTH/2,216-BUF.GETBITMAPHEIGHT/2
BUF.DELETEBITMAP
BUF.CREATEBITMAP 640,432
BUF.SETFORECOLOR RGB(0,0,&HC0)

'ﾊｲｽｺｱ読込み
S$="IROIRO.SCO"
A=0: on error goto *ERR: open S$ for input as #1
  for I=0 to P_ALL: input#1,HI(I): next: close #1: on error goto 0
if A then for I=0 to P_ALL: HI(I)=900000: next: HI_SAVE   '読込み失敗時、初期化

'設定ﾌｧｲﾙ読込み
A=0: on error goto *ERR: open"IROIRO.TXT" for input as #1
  for I=0 to 4: input#1,A$,SND$(I): next: close #1: on error goto 0
if A then                                                    '読込み失敗
時、初期化
  restore *SOUND
  open "IROIRO.TXT" for create as #1: read A$
    for I=0 to 4: read B$,SND$(I): SND$(I)=A$+SND$(I)
      print#1,B$;",";chr$(&H22);SND$(I);chr$(&H22): next
  close #1
endif


'******************************* メイン *******************************
*TITLE
GF=0
MMCLIP 0
while ML=0
  DT=TIMER(100)
  MOUSECHECK MX,MY,ML,A
  BUF.DRAWBITMAP STG,0,0
  BUF.line(0,0)-(639,431),and,1,bf,1: LIGHT 10,320,300,0,1,-1
  if int(rnd(1)*10)<2 then BUF.line(0,0)-(639,431),xor,15,bf,15
  LOGO "★感撃!イロイロ棒★",100,4,4,7
  LOGO "左クリックでスタート",340,3,3,5
  DRAWBITMAP BUF,0,0
wend
MMCLIP 1
MCWAIT

'PLAYER SELECT
GF=1: for I=0 to P_ALL: F(I)=0: G(I)=0: next
do
  DT=TIMER(100)
  MOUSECHECK MX,MY,ML,MR
  P=-1
  for I=0 to P_ALL
```

★6809は懐かしかったです。（大阪市・玉出のタマ36歳）……【**編**：1月号から続けている「CPUの歴史」は人気が高いんですよ。**つぐ美**：でも連載は，今月が最後なんですよね。**編**：趣向を凝らして，またコンピュータの歴史を学ぶ記事やっていきますので，お楽しみに。】

```
  if (MX-X(I))^2+(MY-Y(I))^2<1024 then P=I: exit
 next
 BUF.line(0,0)-(639,431),,0,bf,0
 if P=-1 then LIGHT 3,MX,MY,2,-.5,4 else LIGHT 10,X(P),Y(P),0,1,-1
 D=SPS&(SND$(1+(P=-1)),1)
 for I=0 to P_ALL
  BUF.circle(X(I),Y(I)),32,15,,,,f,,0
  if P=I then PLAYER I,MX,MY,F(I),G(I) else PLAYER I,X(I),Y(I),F(I),G(I)
 next
 for I=0 to 1: LOGO MES$(P+1,I+1),160+I*50,3-I,3-I,7+2*I: next
 if P>-1 then
  LOGO "左右同時クリックで決定",250,2,2,5
  LOGO "BestTime:"+GAMETIME$(HI(P)),290,2,2,13
 endif
 DRAWBITMAP BUF,0,0
 if asc(KB$)=27 then *TITLE
loop while ML=0 or MR=0 or P=-1
D=SPS&(SND$(2),1)
MCWAIT

*START
GF=2: T=0: DT=TIMER(1): F(P)=0: G(P)=0
do
 DT=TIMER(1)
 MOUSECHECK MX,MY,ML,MR
 BUF.DRAWBITMAP STG,0,0
 PLAYER P,MX,MY,F(P),G(P)
 LOGO "青いエリアを左クリックせよ",64-280*(MY<216),2,2,5
 LOGO GAMETIME$(T¥10),-400*(MY<216),2,2,13
 A=(COL(STG.GETPIXEL(MX,MY),&H80)=1) and (COL(HITCHECK(MX,MY),&HC0)<4)
 DRAWBITMAP BUF,0,0
 if asc(KB$)=27 then *TITLE
loop while ML=0 or A=0

'MAIN
GF=3
while T<6000000 and GF=3
 DT=TIMER(1): T=T+DT: if T>6000000 then T=6000000
 MOUSECHECK MX,MY,ML,MR
 BUF.DRAWBITMAP STG,0,0
 PLAYER P,MX,MY,F(P),G(P)
 LOGO GAMETIME$(T¥10),-400*(MY<216),2,2,13
 A=COL(HITCHECK(MX,MY),&HC0)
 if A>3 then GF=4                                                   'ミス
 if COL(STG.GETPIXEL(MX,MY),&H80)=2 and (A and 2) then GF=5         'クリアー
 DRAWBITMAP BUF,0,0
 if asc(KB$)=27 then *TITLE
wend

'CLEAR or GAMEOVER
GF=GF+1
MMCLIP 0
D=SPS&(SND$(GF-2),1)
if GF=6 and T¥10<HI(P) then GF=7: HI(P)=T¥10: HI_SAVE
DT=0: A=-15*(GF=4): D=-272*(MY<216)
while A<20
 A=A+TIMER(100)/50: if A>20 then A=20
 BUF.DRAWBITMAP STG,0,0
 if GF>4 and A<20 then
  B=-(A>6)-13*(A>14): BUF.line(0,0)-(639,431),and,B,bf
  B=1+(abs(10-A)-10)*(GF=5): LIGHT 2*B,MX,MY,B,(GF=5),B
 endif
 PLAYER P,MX,MY,F(P),G(P)
 if A>10 then
  BUF.line(0,D)-step(640,160),and,1,bf,1
  LOGO COM$(GF-4),D+24,4,4,5-6*(GF>5)
  LOGO "左ｸﾘｯｸでﾀｲﾄﾙへ　右ｸﾘｯｸで再挑戦",D+100,2,2,7
 endif
 LOGO GAMETIME$(T¥10),-400*(MY<216),2,2,13
 DRAWBITMAP BUF,0,0
wend
MCWAIT
A=0: while A=0: MOUSECHECK MX,MY,ML,MR: A=ML-2*MR: wend
MMCLIP 1
MCWAIT
if A<0 then *TITLE else *START

'ｴﾗｰﾄﾗｯﾌﾟ
*ERR: A=err: resume next


'************************** サブルーチン・関数 **************************
'ﾊｲｽｺｱ保存                          S$にﾊｲｽｺｱを保存する
sub HI_SAVE()
shared HI()
open S$ for create as #1: for I=0 to P_ALL: print#1,HI(I): next: close #1
end sub


'稲光                               S:光倍率,X,Y:中心,W:太さ,WD:太さ変化,F:ﾌﾗｸﾞ[-1:無
限,0=<:段階指定]
sub LIGHT(S,X,Y,W,WD,F)
A=S-int(rnd(1)*3)*(S>0)
BUF.SETDRAWWIDTH 4*A
BUF.circle(X,Y),13*A,2,,,,,or
BUF.circle(X,Y),9*A,,,,,f,or,3
BUF.SETDRAWWIDTH A
BUF.circle(X,Y),5*A,11,,,,f,or,15
BUF.circle(X,Y),5.5*A,10,,,,,or
for I=0 to 11
 INA X,Y,I/6*PI,W,WD,F
next
```

★OFコーナーで1行プログラムを募集するのはどうでしょう。（広島県豊田郡・・・13歳）……【影：どんどんやってください。募集していなくても，やりたいこと，おもしろいことがあったら，どしどし投稿してください。というわけで1行プログラム待ってます。】

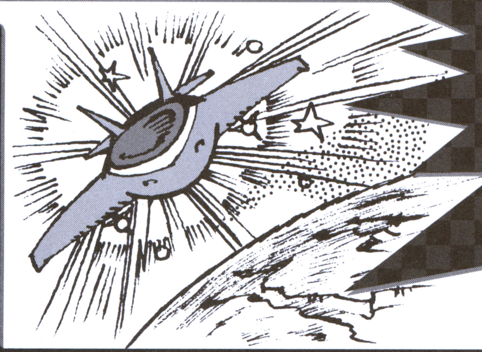
```
BUF.SETDRAWWIDTH 1
end sub

'稲妻                              X,Y:稲妻発生点,DV:方向,W:太さ,WD:太さ変化,F:ﾌﾗｸﾞ[同
上]
sub INA(byval X,byval Y,byval DV,byval W,byval WD,byval F)
while X>-32 and X<672 and Y>-32 and Y<464 and F
  W=W+WD
  A=DV+(rnd(1)*.6-.3)*PI
  L=int(rnd(1)*8)*10+20
  F=F-1
  X0=X: Y0=Y
  X=X0+L*cos(A): Y=Y0+L*sin(A)
  for J=0 to 1
    BUF.SETDRAWWIDTH (3-J)*W
    BUF.line(X0,Y0)-(X,Y),or,11+4*J
  next
wend
end sub

'ロゴ表示                          A$:文字列,Y:表示Y座標,W:横倍率,H:縦倍率,C:色
sub LOGO(A$,Y,W,H,C)
X=320-4*W*len(A$)
for I=-1 to 1 step 2: BUF.symbol(X+I,Y+I),A$,W,H,0: next
BUF.symbol(X,Y),A$,W,H,C
end sub


'ﾏｳｽﾁｪｯｸ                           MX,MY:ﾏｳｽ座標,LC,RC:左右ｸﾘｯｸ...の情報が帰る
sub MOUSECHECK(MX,MY,LC,RC)
MX=mouse(0): MY=mouse(1)
A=(mouse(2,0)=-1 or mouse(11,0)<>0): B=LC-(mouse(6,0)<>0)
C=(mouse(2,1)=-1 or mouse(11,1)<>0): D=RC-(mouse(6,1)<>0)
if LC=0 then LC=A else LC=B
if RC=0 then RC=C else RC=D
KB$=ucase$(inkey$+"_")
end sub

'ﾏｳｽ移動範囲限定/解除              F:ﾌﾗｸﾞ[0:解除,1:画面内に制限]
sub MMCLIP(F)
var WINSIZ as RECT, DUMMY as long
shared MX,MY
if F then
  WINSIZ.X1=GETLEFT+GSM&(32)+1
  WINSIZ.Y1=GETTOP+GSM&(33)+GSM&(31)+1
  WINSIZ.X2=WINSIZ.X1+GETCLIENTWIDTH-1
  WINSIZ.Y2=WINSIZ.Y1+GETCLIENTHEIGHT-1
  A=CC1&(WINSIZ): SETPOINTER MUS
else
  A=CC2&(0): SETMOUSEPOINTER 0
endif
end sub

'ﾏｳｽｸﾘｯｸOFF待ち
sub MCWAIT()
shared ML,MR
do
  MOUSECHECK A,B,ML,MR
loop while ML+MR<>0
end sub

'ﾀｲﾏｰｶｳﾝﾄ                          WT:待機時間(ﾐﾘ秒:0指定禁止),戻り値:実際の待機時間
function TIMER(WT)
A#=GTC_TIMER#: while GTC_TIMER#<A#+WT
  GTC_TIMER#=GTC&: GTC_TIMER#=GTC_TIMER#-(GTC_TIMER#<A#)*&HFFFFFFFF
wend: TIMER=GTC_TIMER#-A#
end function

'時間文字列                        T:ﾀｲﾑ,戻り値:時間文字列
function GAMETIME$(T)
if T>600000 then GAMETIME$="-:--.--": exit function
A=T¥6000: B=100.001+(T-A*6000)/100
GAMETIME$=trim$(str$(A))+":"+mid$(str$(B),3,5)
end function

'ｶﾗｰ変換                           C:RGB値,L:輝度(0-255),戻り値:ｶﾗｰｺｰﾄﾞ0-7(L輝度以上を返
す)
function COL(C,L)
A$=right$(hex$(C),6)
A$=string$(6-len(A$),"0")+A$
R=val("&h"+mid$(A$,5,2))
G=val("&h"+mid$(A$,3,2))
B=val("&h"+mid$(A$,1,2))
COL=-(B>=L)-2*(R>=L)-4*(G>=L)
end function


'ﾋｯﾄﾁｪｯｸ                           X,Y:ﾌﾟﾚｲﾔｰ座標,戻り値:ﾌﾟﾚｲﾔｰと重なったﾄﾞｯﾄのｶﾗｰ論理和
function HITCHECK(X,Y)
shared PL()
PL(0).SETDRAWMODE 9
PL(0).BITBLT 0,0,64,64,STG,X-32,Y-32
for I=5 to 0 step -1: A=2^I: for J=0 to 3: B=I and 1
  PL(B).SETDRAWMODE 15+2*(J=0)
  PL(B).BITBLT 0,0,A,A,PL(1-B),A*(J¥2),A*(J and 1)
next: next
HITCHECK=PL(0).GETPIXEL(0,0)
end function


'ﾌﾟﾚｲﾔｰ描画                        N:棒の種類,X,Y:ﾌﾟﾚｲﾔｰ座標,F,G:情報(各ｷｬﾗ毎に意味が変
わる)
sub PLAYER(N,X,Y,F as double,G as double)
```

★これからもPCハードウエアの特集を続けてください。(大阪府茨木市・J+1 29歳)……**【影：ソフトとハードは，私とつぐ美ちゃんのように，切っても切れない間柄。もちろん，ハードの企画も続けていきます。つぐ美：影さんと私が何ですって？影：まあまあ。】**

```
shared PL(),PX(),PY(),MES$(),X(),Y(), ML,MR,DT,GF,P
PL(0).SETDRAWMODE 13
PL(0).line(0,0)-(63,63),,0,bf,0
A$=MES$(N+1,0)
select case A$
case "普通","極太"                    'F,G:未使用
  PL(0).circle(32,32),5+(A$="普通")*2,15,,,,f
case "風船"                           'F:伸縮ｶｳﾝﾀ,G:未使用
  F=F+DT/100
  PL(0).circle(32,32),abs(F mod 16)-8),15,,,,f
case "プロペラ"                       'F:回転ｶｳﾝﾀ,G:未使用
  F=F+DT/2000*PI*(1+2*(ML or MR))
  A=8*sin(F): B=8*cos(F)
  PL(0).SETDRAWWIDTH 2
  PL(0).line(32+A,32+B)-(32-A,32-B),,15
  PL(0).SETDRAWWIDTH 1
case "リボン"                         'F,G:未使用、PX()PY()を専用配列として使用
  PL(0).circle(32,32),3,15,,,,f
  PL(0).SETDRAWWIDTH 2
  PX(0)=X+32: PY(0)=Y+32
  for I=1 to 8
    C=16*DT/1000
    if I=1 then PY(I)=PY(I)+C else PX(I)=PX(I)+C*A: PY(I)=PY(I)+C*B
    A=PX(I)-PX(I-1): B=PY(I)-PY(I-1): C=sqr(A*A+B*B)+.001: A=A/C: B=B/C
    if C>4 then PX(I)=PX(I-1)+4*A: PY(I)=PY(I-1)+4*B
    PL(0).line(PX(I-1)-X,PY(I-1)-Y)-(PX(I)-X,PY(I)-Y),,15
  next
  PL(0).SETDRAWWIDTH 1
case "二極"                           'F:角度,G:未使用
  F=F-(ML-MR)*DT/3000*PI
  A=16*sin(F): B=16*cos(F)
  PL(0).circle(32+A,32+B),2,15,,,,f
  PL(0).circle(32-A,32-B),2,15,,,,f
case "ひし形"                         'F:扁平度合[-:縦伸び,0:正方形,+:横伸び],G:未使用
  F=F+(ML-MR)*DT/100: if F>6 then F=6 else if F<-6 then F=-6
  A=F+7: B=14-A
  PL(0).connect(32-A,32)-step(A,-B)-step(A,B)-step(-A,B),15,,f,15
case "点滅"                           'F:点滅ｶｳﾝﾀ,G:未使用
  F=F+DT/400
  PL(0).circle(32,32),-6*(F mod 4<3),15,,,,f
case "振り子"                         'F:振り子X座標,G:未使用
  F=F+DT/400: A=12*sin(F): B=20: C=44: D=32
  if P<>N then D=D-A else B=B+A: C=C+A: X=X+(GF<4)*A
  PL(0).line(B,30)-(C,34),,5,bf,5
  PL(0).circle(D,32),2,15,,,,f
case "ミラー"                         'F,G:未使用、GFで反転基準点を変更
  if GF=1 then X=X(N)*2-X else if GF=2 or GF=3 then X=639-X
  PL(0).circle(32,32),2,15,,,,f
case "ロケット"                       'F:Y座標,G:Y速度
  if GF=2 or P<>N then
    G=0
  else
    A=G: Y=F: G=G+(2*(ML or MR)+1)*DT/500: if abs(G)>1 then G=sgn(G)
    Y=Y+(A+G)*DT/80
    if GF=1 and abs(Y-Y(N))>32 then Y=Y(N)+sgn(Y-Y(N))*32
  endif
  F=Y: A=-(ML or MR)*(rnd(1)*8+8)
  PL(0).connect(32,34)-step(-A,A)-step(A,-A/2)-step(A,A/2),4,,f,5
  PL(0).circle(32,34),2,15,2,.5,0,f
case "ドライブ"                       'F:XY座標,G:角度
  A=ML-MR
  G=G-A*DT/1000*PI: A=2*sin(G): B=2*cos(G)
  if P=N and GF<>2 then
    X=int(F)/1000-100: Y=(F-int(F))*1000-100
    X=X+A*DT/100: Y=Y+B*DT/100
  endif
  F=clng((X+100)*1000)+(Y+100)/1000
  PL(0).connect(32-B,32+A)-step(4*A,4*B)-step(2*B,-2*A)-step(-4*A,-4*B),15,,f,15
  PL(0).connect(32-B+2*A,32+A+2*B)-step(A,B)-step(2*B,-2*A)-step(-A,-B),15,,f,0
end select
BUF.SETDRAWMODE 7
BUF.DRAWBITMAP PL(0),X-32,Y-32
BUF.SETDRAWMODE 13
end sub


'****各種データ****
*PL
data 11                              '棒の種類-1
data "NoSel","棒を選択せよ","雷を使いたい棒に重ねよう"
data "普通","入門","イライラしない?標準サイズ"
data "極太","上級","道幅ギリギリの高難度"
data "風船","伸縮","太く細く…タイミングを計れ"
data "プロペラ","回転","クリックすると逆回転"
data "リボン","華麗","止まると帯がたれてくる"
data "二極","慎重","左/右クリックで左/右に回転"
data "ひし形","変形","左/右クリックで縦/横に伸縮"
data "点滅","困惑","消えると無敵だが見失うぞ"
data "振り子","激動","左右に揺れるので注意"
data "ミラー","混乱","左右逆に動いてしまう"
data "ロケット","発進","クリックで上昇、離すと下降"
data "ドライブ","爆走","左/右クリックで左折/右折"


*SOUND
data "C:¥Program Files¥Plus!¥Themes¥"  'ｻｳﾝﾄﾞﾌｧｲﾙがあるﾌｫﾙﾀﾞ
data "ｶﾐﾅﾘ音  ","コンピュータの内側 - 最大化.wav"
data "PL選択音","コンピュータの内側 - 感嘆符.wav"
data "PL決定音","レオナルド ダ ビンチ - 起動.wav"
data "ｸﾗｯｼｭ音 ","コンピュータの内側 - Windows の終了.wav"
data "ｸﾘｱ音   ","レオナルド ダ ビンチ - アスタリスク.wav"
```

★テスト中にもかかわらず，PCをやっています。どうしたら止められますでしょうか（←自制心がないだけですよね）。（愛知県名古屋市・**子興@しよ16歳**）……【**影**：ま，気分転換も大事だから，いいんじゃない。**編**：で，テストはどうだったんでしょうね。】

# Windows(HSP V2.55)用



HSPによる3Dシューティング

# Defense-line（ディフェンスライン）

くろすけ（34歳）

## ゲームの説明

次々と向かってくる敵機を撃ちまくり，防衛ラインを守ってください。遠くで撃破するほど高得点になるので，ハイスコアを競って頑張ってね。

100機以上の敵機に防衛ラインを突破されるとゲームオーバーです。

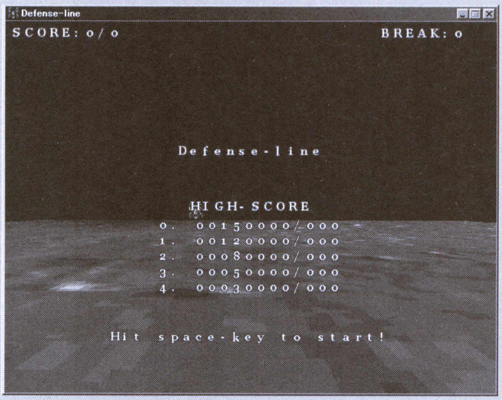
## 操作方法

カーソルキーで左右に移動し，スペースキーで弾を撃ちます。

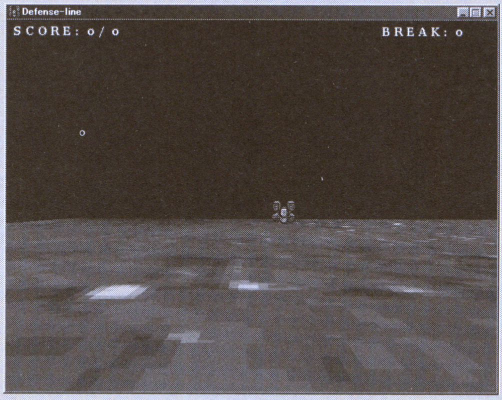
## プログラムの入力方法

まず，絵と音のデータを作成しましょう。

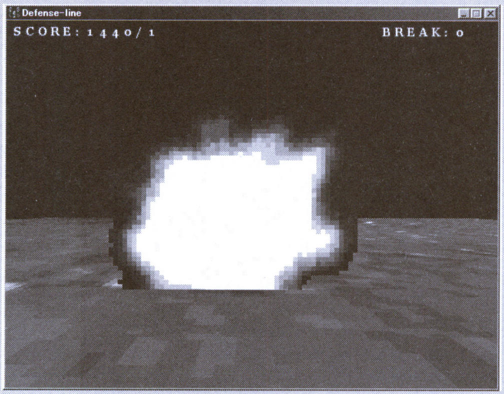
画像データは，**第1表**を参考にペイントなどで作成してください。



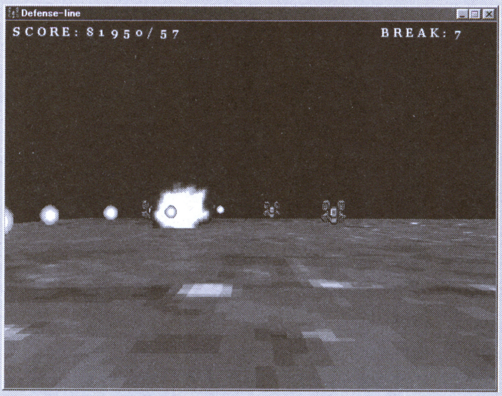

《写真1》奥から攻めてくる敵機を倒す




《写真2》画面奥からの敵機を発見したら




《写真3》まよわず弾を連射。敵機を撃破しよう




《写真4》爽快感のあるシューティングが楽しめる

★好きな子と学校で慣れっぽく話をするには，どうしたらいいでしょうか？（北海道天塩町・**バーテックス16歳**）……【**影**：ねえ，つぐ美ちゃん，昼ご飯，何食べに行こうか。ラーメン？中華？**つぐ美**：そうね，パスタがいいわね，編さん。**影**：うまくいかないものだ。】

効果音データは，以下のような対応になっています。

start.wav：ゲーム開始！
shot.wav：ショット
explos.wav：敵の爆発
end.wav：ゲームオーバー

最後に**リスト**を入力し，コンパイル，実行すれば，とりあえず遊べます。

EXEファイルの作り方については，HSPのマニュアルを見てね。

## プログラムについて

HGIMGの3D機能としては，最も単純なプレートモデルだけを利用したゲームです。

DirectXモードによる描画を試したい場合は，最初の「#include "hgimg.as"」の部分を「#include "hgimgx.as"」としてください。

## 最後に一言

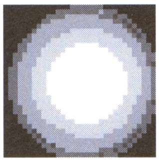
最近，夢でみた光景をなんとなくゲームにしてみました。作者のハイスコアは，510,460点，355機撃墜。超えてみよ！

## CHECKER FLAG

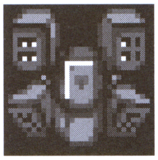
**影**：スタンダードな3Dゲームだけど，やり始めるとかなりハマル。連射機能はモチロン，スムーズな動きや敵の爆破効果などがしっかりしているので，プレイしていて爽快感があるなぁ。
**編**：この作品から，HSPでの3Dゲームの流れを学んでほしいですね。
**Dr.D**：このリストの長さで3Dゲームが実現できるのがHSPの強みじゃな。また，コメントもしっかり記述されているので，改造も行いやすい。まずは，変数の値を変更するなどして，敵の移動スピードやゲームオーバー条件などを変更してみてはどうかな。

《第1表》画像ファイルの内容

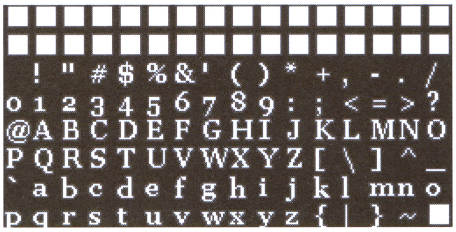
| ファイル名 | 大きさ |
|---|---|
| enemy.bmp | 32×32ドット |
| explos.bmp | 80×960ドット<br>(80×80ドットの爆発パターン×12) |
| floor.bmp | 256×256ドット |
| shot.bmp | 32×32ドット |
| font.bmp | 256×128ドット<br>(1文字16×16ドット) |



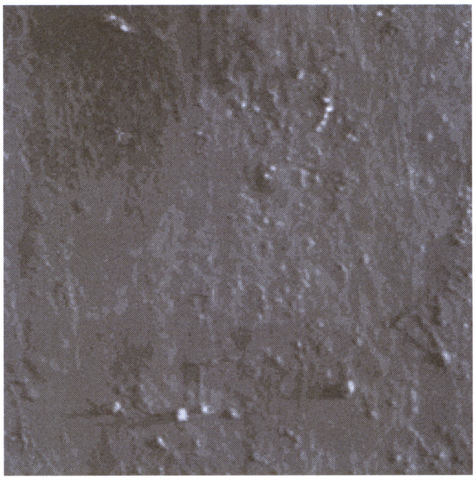
shot.bmp



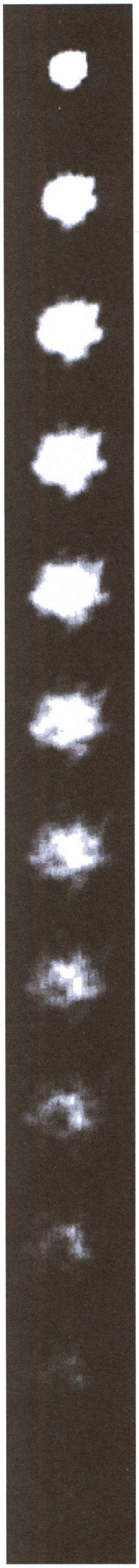
enemy.bmpの画像




fom.bmpの画像



floor.bmpの画像



explos.bmpの画像



入力方法は47ページをご覧ください。

★私の意見は，ExcelなどのVBAプログラミング記事は，初心者には大変良いと思います。弊社の「日本語OS」もよろしく！！（京都市・PAPPY33歳）……【編：弊社？日本語OS？どこの会社にお勤めなんでしょう。または自分で会社やってるとか？気になります。】

## 《リスト》Defense-lineのプログラム・リスト

```
; Defense-line

#include "hgimg.as"

        xs=640 :ys=480
        screen 0,xs,ys,1,dispx-xs/2,dispy-ys/2
        title "Defense-line"

        hgini                                           ; HGIMGの初期化
; ----------------------------------------------------------------------
        setfont 16,16,16,1 :texload "font.bmp"          ; ﾌｫﾝﾄﾃｸｽﾁｬ登録
; ----------------------------------------------------------------------
        setsizef 256.0f,256.0f                          ; ﾌﾟﾚｰﾄXYｻｲｽﾞ
        setuv 0,0,255,255                       ; ﾃｸｽﾁｬUV座標
        addmesh mdfr,16,16                              ; ﾌﾟﾚｰﾄﾓﾃﾞﾙ作成
        texload "floor.bmp"                             ; 地面ﾃｸｽﾁｬ読込み
; ----------------------------------------------------------------------
        setsizef 4.0f,4.0f                              ; ﾌﾟﾚｰﾄXYｻｲｽﾞ
        setuv 0,0,31,31                                 ; ﾃｸｽﾁｬUV座標
        addplate mdsh,1                                 ; ﾌﾟﾚｰﾄﾓﾃﾞﾙ作成
        texload "shot.bmp"                              ; 弾ﾃｸｽﾁｬ読込み
; ----------------------------------------------------------------------
        setsizef 8.0f,8.0f                              ; ﾌﾟﾚｰﾄXYｻｲｽﾞ
        setuv 0,0,31,31                                 ; ﾃｸｽﾁｬUV座標
        addplate mden,1                                 ; ﾌﾟﾚｰﾄﾓﾃﾞﾙ作成
        texload "enemy.bmp"                             ; 敵ﾃｸｽﾁｬ読込み
; ----------------------------------------------------------------------
        setsizef 32.0f,32.0f                            ; ﾌﾟﾚｰﾄXYｻｲｽﾞ
        setuv 0,0,79,79                                 ; ﾃｸｽﾁｬUV座標
        addplate mdex,1                                 ; ﾌﾟﾚｰﾄﾓﾃﾞﾙ作成
        texload "explos.bmp"                            ; 爆発ﾃｸｽﾁｬ読込み
        uvanim mdex,5,12,1                              ; ｱﾆﾒｰｼｮﾝ設定
; ----------------------------------------------------------------------
        sndload "start.wav"                             ; ｹﾞｰﾑｽﾀｰﾄの音
        sndload "shot.wav",1                            ; 弾の発射音
        sndload "explos.wav",2                          ; 敵の爆発音
        sndload "end.wav",3                             ; ｹﾞｰﾑｵｰﾊﾞｰの音

        setborder 256.0f,32.0f,320.0f                   ; ｹﾞｰﾑ空間設定

        regobj obfr,mdfr                        ; 地面ｵﾌﾞｼﾞｪｸﾄを作成
        selang obfr :objsetf1 0,-1.57f                  ;
        selpos obfr :objset3 8,0,0                      ;
*start
        sdim buf,256 :dim hs,8 :dim hn,8                ; ﾊｲｽｺｱ登録
        fn="best5.cnt" :exist fn
        if strsize>0 { bload fn,buf                             ; 記録ﾌｧｲﾙが存在しな
ければ作成する
        } else {
buf="00150000,000¥n00120000,000¥n00080000,000¥n00050000,000¥n00030000,000"
        }
        notesel buf :i=0                                ; ｽｺｱ順位をｿｰﾄ
        repeat 5 :noteget a,cnt                                 ;
                strmid b,a,9,3 :int a :int b                    ;
                if sc>a :hs.i=sc :hn.i=nm :sc=-1 :i+            ;
                hs.i=a :hn.i=b :i+                              ;
        loop
        repeat 5                                        ; ﾊｲｽｺｱ記録ﾌｧｲﾙを保存
                a=hs.cnt :b=hn.cnt :str a,8 :str b,3            ;
                noteadd a+","+b,cnt,1                           ;
        loop                                                    ;
        strlen l,buf :bsave fn,buf,l                            ;

        df=699 :sc=0 :nm=0 :ov=0                        ; 難易度 / ｽｺｱ / 撃墜数 / 突
破数
        edz.0=0.5f :edz.1=0.75f :edz.2=1.0f             ; 敵速度3ﾊﾟﾀｰﾝ

        selcpos :objset3 0,-8,128                       ; 自機(ｶﾒﾗ)座標

        dx=0 :tm=0                                              ; 自機移動量 / ｹﾞｰﾑｵ
ｰﾊﾞｰ処理中ﾀｲﾏｰ
        dm=1 :ez=0                                              ; ﾃﾞﾓ用ﾌﾗｸﾞ / 難易度
軽減ﾌﾗｸﾞ

        gsel 0,1
*main
        if dm=1 { stick ky                                      ; ﾃﾞﾓ
                if ky=16 { snd 0 :dm=0 :findobj 0,2 ;
                        repeat :nextobj o               ; 敵ｵﾌﾞｼﾞｪｸﾄのｸﾘｱ
                                if o=-1 :break                  ;
                                delobj o                ;
                        loop                                    ;
                }
                goto *skp
        }
```

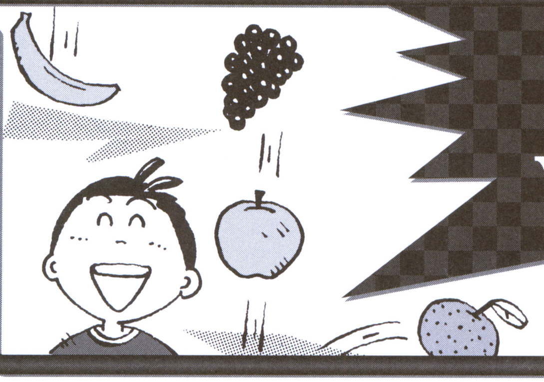
★インタクタコードでBMPファイルを掲載するというのはどうでしょうか。自分で作るのはなかなか難しいので。(茨城県下館市・AMCP22歳) ……【Dr.D：「インタクタコード」とは音声や画像データを紙に印刷する技術。スキャナで読み取ることでPCに表示するんじゃな。】

```
	if tm>0 { tm-                                        ; ゲームオーバー処理中
		if tm=0 :goto *start :else :goto *skp            ;
	}
	if ov>99 { selcpos :objgetfv p                       ; ゲームオーバーの判定
		regobj obex,mdex,OBJ_UVANIM_1SHOT   ;
		selpos obex :fsub p.2,8.0f :objsetfv p           ;
		snd 3 :tm=500                                    ;
	}

	findobj 0,2                                          ; 突破した敵の判定
	repeat :nextobj o :if o=-1 :break             ;
		selpos o :objgetfv p :f2i z,p.2                  ;
		if z>132 :delobj o :ov+                          ;
	loop                                                 ;

	findobj 0,1                                          ; 弾が敵に当たって
るか
	repeat :nextobj o :if o=-1 :break             ;
		getcoli f,o,5.0f                          ;
		if f>-1 { selpos f :objgetfv p                   ;
			delobj o :delobj f
			regobj obex,mdex,OBJ_UVANIM_1SHOT
			selpos obex :objsetfv p :snd 2
			f2i a,p.2 :sc=sc+(144-a*10) :nm+
			if df>99 :df-=5
		}
	loop

	if (nm>80)&(ez=0) :ez=1 :df+=250              ; 80機以上突破されたら難易度
を軽減

	stick ky,15
	if (ky&1>0)&(dx>-150) :dx-=5                         ; 自機左右加速
	if (ky&4>0)&(dx<150) :dx+=5                          ;
	if (ky&5)=0 :dx=dx*12/13                      ; キー押してなければ減速

	str2f f,""+dx :fdiv f,100.0f                         ; 移動量 dx → f 少
数化

	selcpos :objgetfv po                                 ;
	fadd po.0,f                                          ; 自機を移動
	fvmax po,128.0f,-8.0f,128.0f                         ;
	fvmin po,-128.0f,-8.0f,128.0f                        ;
	objsetfv po                                          ;

	t=t-(t>0)
	if (ky&16>0)&(t=0) { t=13 :snd 1
		regobj obsh,mdsh,OBJ_MOVE|OBJ_BORDER         ; 弾オブジェクト発射(作
成)
		setcoli obsh,1,2                          ;
		seldir obsh :objset3 0,0,-2                      ;
		selpos obsh :fadd po.1,2.0f :objsetfv po
	}
*skp
	rnd r,df
	if r<3 { regobj oben,mden,OBJ_MOVE|OBJ_BORDER        ; 敵オブジェクトを発生
(作成)
		setcoli oben,2                                   ;
		rnd x,240 :rnd f,3 :x=x-120 :f-                  ;
		str2f f,""+f :fdiv f,10.0f                ;
		selpos oben :objset3 x,-6,-144                   ;
		seldir oben :objsetf3 f,0.0f,edz.r ;
	}

	hgdraw                                               ; 全オブジェクト描画処理

	if dm { fprt "Defense-line",224,160           ; タイトル表示(デモ)
		fprt "HIGH-SCORE",240,232                 ;
		repeat 5
			a=hs.cnt :b=hn.cnt :str a,8 :str b,3
			fprt ""+cnt+". "+a+"/"+b,200,20*cnt+256
		loop
		fprt "Hit space-key to start!",136,400
	}

	fprt "SCORE:"+sc+"/"+nm,8,8                          ; スコア / 撃墜数
	fprt "BREAK:"+ov,488,8                               ; 敵の突破数

	if (tm<380)&(tm>0) :fprt "GAME OVER",248,232; ゲームオーバー処理中の表示

	hgsync 10                                            ; ウェイト

	goto *main
```



★投稿プログラムコーナー，おもしろかったです。自分もいつか投稿したいです。それまで待っていてください。（群馬県邑楽郡・factor16歳）……【影：さて，いつまで待ちましょうか。編：影さんのデートの場合，待ちぼうけが多いそうですが？影：ウーっ。】

# Javaアプレット用



野菜を並べる落ち物パズル

# vegetable3（ベジタブルスリー）

長谷川　誠（37歳）

## ゲームの内容

さて，このゲームなんですが，「MSX・FAN」という雑誌があって，こんな感じのゲームがあったような気がします。それを記憶をたよりに作ったんですが，移植にあたるのかどうか…？

落ちてくる5種類の野菜を，以下のようにすると得点になります。

・同じ種類の野菜を3個重ねる(**写真2**)。

・5種類の野菜を重ねる(**写真3**)。

・スペシャルの組み合わせに重ねる(画面右に表示されている。並べ方は順不同)。

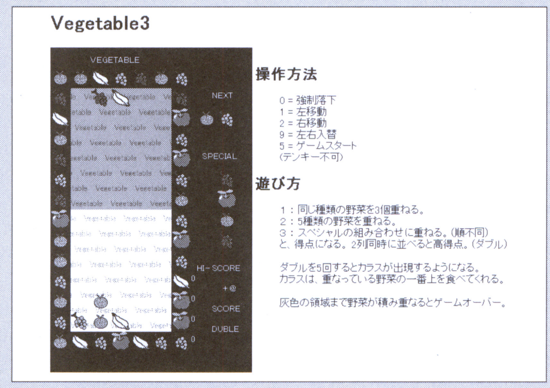
2列同時に並べる(ダブル)と高得点。ダブルを5回すると一番上の野菜を食べてくれるカラスが出現するようになります。

灰色の領域まで野菜が積み重なるとゲームオーバー(**写真4**)。

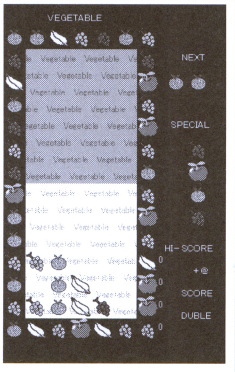
## 操作方法

まず，vegetable3.html(**リスト2**)をInternetExplorerで読み込んで，Javaアプレットを立ち上げてください。

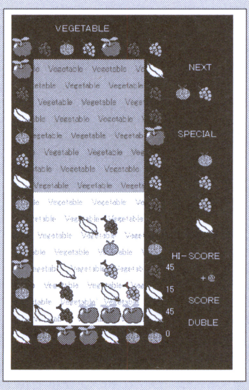
操作方法は，[0]キーで強制落下，[9]キーで左右入れ替え，[1]キーで左移動，[2]キーで右移動，[5]キーでゲーム開始です。キー入力を受け付けないときは，アプレット内をマウスでクリックしてみてくだ



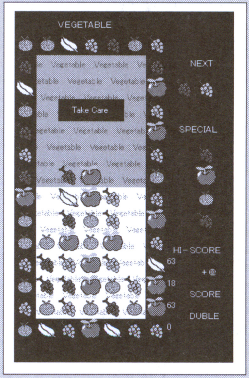

《写真1》野菜を題材にした落ち物パズル




《写真2》同じ野菜を3個重ねると消せる




《写真3》5種類の野菜を重ねても消去できる




《写真4》灰色の領域に入るとダメ。画面上ではない点に注意



★セガの8ビット機もZ80ですよね。セガのメガドライブもMC68000ですよね。（長野県北安曇郡・伊藤　優18歳）……【Dr.D：いろんなところでCPUの歴史記事が反響をよんでおるようじゃな。編：ある学校の先生は，授業のテキストの参考にしたいと言ってました。】

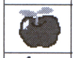
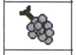
**《第１表》変数表**

| | |
|---|---|
| score | 得点 |
| scup | 加算得点 |
| scup2 | 加算得点表示用 |
| hisc | 最高得点 |
| lvr | 役成立回数 |
| spcl | スペシャル役判定用 |
| yakudbl | ダブル成立回数 |
| hantei | 役判定用 |
| rakkasp | 落下時間(rakkasp×10ミリ秒(0.01秒)) |
| ct | 落下時間カウンタ |
| spcflg | スペシャル成立フラグ(１：成立，０：不成立) |
| gamesw | 描画ルーチン切替用(１：下絵、画像読込待ち，９：ゲーム画面) |
| numkey | キー入力用 |
| keyflg | 自然落下時，キー入力拒否フラグ(true：受付) |
| keyflg1 | 横移動時，キー連続入力防止フラグ(true：受付) |
| keyflg2 | 強制落下時，キー連続入力防止フラグ(true：受付) |
| gameov | ゲーム状況(true：ゲームオーバー) |
| vw,vh | 仮想画面の幅と高さ |
| vx,vy | 左野菜位置(右野菜はvx＋１，vy) |
| d | アプレット表示領域 |
| offs | オフスクリーン用(offs1，offs2も同様) |
| grf | オフスクリーン用(grf1，grf2も同様) |
| kicker | runメソッドのメインルーチンをループさせるためのスレッド変数 |
| mt | MediaTrackerクラス利用時，必要みたい |
| keisan[] | 野菜別役判定評価値 |
| board[] | 仮想画面 |
| nextvgtbl | 次落下野菜 |
| spclvgtbl | スペシャル役組合せ保持用 |
| pict[] | 野菜画像データ |

**《リスト１》vegetable3.javaのプログラム・リスト**

```
// 2001年12月10日製作開始
// 長谷川　誠

import java.applet.Applet;
import java.awt.*;
import java.util.*;
import java.lang.*;
import java.awt.event.*;

public class vegetable3 extends java.applet.Applet implements Runnable,KeyListener
{
        int score,scup,scup2,hisc;  //−得点、加算得点、加算得点表示用、最高得点
        int lvr,yakudbl,hantei,spcl; //−役成立回数，ダブル成立回数，役判定用，スペシャル判定用
        int rakkasp,ct,spcflg;       //−落下時間、カウンタ、スペシャル成立フラグ
        int gamensw = 1;     //−描画ルーチン切替え用
        int numkey = -1;     //−キー入力用
```

**《第２表》画像ファイル**

| 画像 | | ファイル名 |
|---|---|---|
| | それぞれ32×32ドット | img1.gif |
| | | img2.gif |
| | | img3.gif |
|  | | img4.gif |
|  | | img5.gif |
| | | img6.gif |

さい。なお，数字入力の操作ではありますが，テンキーは使用できません。

## プログラムの入力

画像ファイルを作成する必要があります。32ドット×32ドットの透過Gifで５種類の野菜(img1.gif～img5.gif)とカラス(img6.gif)を作ってください。私は付属ソフトの「ペイント」で作画し，「フォトエディタ」で変換しました(両方とも初めて使いました)。

次にvegetable3.java(**リスト１**)をテキストエディタで入力します。入力後，セーブしてからコンパイルして，classファイルを作ってください。

vegetable3.html(**リスト２**)は，Javaアプレットをブラウザ上で呼び出すためのものです。テキストエディタ(「メモ帳」など)で入力してセーブしてください。

## プログラムについて

理解していないけれど，参考にしたものをそのまま使っているところもあるので，要らない，あるいはおかしな部分があるかもしれません。

役が成立したとき点滅しながらひとつずつ消えるようにしたかったけれど，描画メソッドが自分の思ったとおりに動かなくて，結局できませんでした。

## あとがき

８年ぐらい前にMSXというパソコンでプログラミングしてたんですけど，去年の７月に久しぶりにパソコンを買ってみたら，まったく違うものになってて…。時代の流れを感じました。Javaは昨年９月から勉強しはじめたんですが，Javaは今後どうなっちゃうんでしょうね。

手軽に遊べるゲームですので，気がむいたら遊んでみてください。私の最高点は130,000点です。

### CHECKER FLAG

**つぐ美**：ブラウザで遊べるお手軽感がいいわね。Java言語なので，画面構成を工夫すれば，iアプリへの移植も行えそうですね。
**編**：また，このゲームを元に，各自でいろいろな改造を行って，よりオリジナリティのあるゲームに仕上げてみるのもありですね。
**影**：典型的な落ち物パズルだけど，野菜の消去条件に工夫がある。
**Dr.D**：その消去判定(役判定)は，1,10と桁をずらした値を格納してあるkeisan配列を用いて，そのkeisan内の値を加算した結果から判定するという，チョット面白い方法を用いておる。この方法を用いることで，５種類の野菜組み合わせ判定を手軽に行っておるのが特徴じゃな。


入力方法は50ページをご覧ください。

★各コーナーでメールでの意見募集とかすると良いよ。CMMよ，いま一度！つかちーよ，いま一度…。(福井県武生市・**つかち～。17歳**)……【**編**：メールでの意見・感想は随時受け付けています。webmas@basicmagazine.netまで，お寄せください。よろしく！】

```
int vw=6,vh=13;      //-—仮想画面の幅と高さ
int vx,vy,x,y;       //-—左画面野菜位置,ループ用

int keisan[] = new int[10];    //-—役判定用
int board[][] = new int[vw+1][vh+1];    //-—仮想画面
int nextvgtbl[] = new int[2];  //-—次に落ちてくる野菜
int spclvgtbl[] = new int[4];  //-—スペシャル役組合せ保持用
Image pict[] = new Image[10];  //-—野菜画像データ

Thread kicker = null;          //-—メインルーチンをループさせるためのスレッド変数
MediaTracker mt;     //-—画像を読み込むまで待つための変数

Dimension d;         //-—表示領域
Image offs,offs1,offs2;        //-—オフスクリーン
Graphics grf,grf1,grf2;

boolean keyflg = false,keyflg1 = false,keyflg2 = false;      //-—key入力受付フラグ
boolean gameov = true;         //-—ゲーム状況
public void init()  {          //=================

        mt = new MediaTracker(this);

        /* —-イメージファイルの読み込み—- */
        for(int i=0 ; i<10 ; i++)       {
                String fileName = getParameter("imagefile."+i);
                if(fileName != null){
                        pict[i] = getImage(getDocumentBase(),fileName);
                        mt.addImage(pict[i],0);//—-ID番号0でイメージを登録
                }
        }

        /*-—オフスクリーンの設定—-*/
        d = getSize();         //-—表示領域の確保
        offs = createImage(d.width,d.height);
        grf = offs.getGraphics();
        offs1 = createImage(384,24);
        grf1 = offs1.getGraphics();
        offs2 = createImage(32*(vw-1),32*(vh-1));
        grf2 = offs2.getGraphics();

        /*-—キーリスナーとして自分自身を登録---*/
        addKeyListener(this);
        requestFocus();        //-—キー入力フォーカス

        /*-—変数と初期画面設定—-*/
        keisan[1] = 1 ; keisan[2] = 10 ; keisan[3] = 100 ;
        keisan[4] = 1000 ; keisan[5] = 10000;
        gameinit();
        nextvset();
        spclv();
}

public void paint(Graphics g)  {          //=============
        update(g);
}
public void update(Graphics g) {
```

```
                int i,j;

                switch(gamensw)      {
                        case 1:   //—-play画面のオフスクリーン作成
                                  for(j=0 ; j<=1 ; j++)          {
                                          if(j==0)  {
                                                  grf1.setColor(Color.lightGray);
                                                  grf1.fillRect(0,0,384,24);
                                                  grf1.setColor(Color.gray);
                                          } else {
                                                  grf1.setColor(Color.white);
                                                  grf1.fillRect(0,0,384,24);
                                                  grf1.setColor(Color.lightGray);
                                          }
                                          grf1.drawString("Vegetable   Vegetable
Vegetable   Vegetable   Vegetable   Vegetable",0,18);

                                          for(i=0 ; i<=7 ; i++)          {
                                          grf2.drawImage(offs1,i*24168,i*24+j*192,this);
                                          }
                                  }

                                  /*—-画像読み込み時のメッセージ表示—-*/
                                  g.setColor(Color.black);
                                  g.fillRect(0,0,d.width,d.height);
                                  g.setColor(Color.white);
                                  g.drawString("Please wait.",d.width/4,d.height/2);

                                  gamensw = 9;
                                  break;

                        case 9:   //—-バックを塗る
                                  grf.setColor(Color.black);
                                  grf.fillRect(0,0,d.width,d.height);

                                  /*—-next野菜の表示-—*/
                                  grf.setColor(Color.white);
                                  grf.drawString("ＮＥＸＴ",256,80);
                                  grf.drawImage(pict[nextvgtbl[0]],232,96,this);
                                  grf.drawImage(pict[nextvgtbl[1]],264,96,this);

                                  /*—-specialの組み合わせの表示-—*/
                                  grf.drawString("ＳＰＥＣＩＡＬ",240,176);
                                  for(i=0 ; i<4 ; i++){
                                          grf.drawImage(pict[spclvgtbl[i]],264,192+i*32,
this);
                                  }

                                  /*—-play画面の表示-—*/
                                  grf.drawImage(offs2,32,64,this);

                                  for(i=0 ; i<=vw ; i++){
                                    for(j=0 ; j<=vh ; j++){
                                      if(board[i][j] > 0){
                                      grf.drawImage(pict[board[i][j]],i*32,j*32+32,this);
                                      }
                                    }
```



★影さんって，ザクみたいで，かっこいいです。（群馬県高崎市・いせるはーさのたびびと22歳）……【Dr.D：ザク？何じゃ？影：「機動戦士ガンダム」に登場するモビルスーツですよ。なんにしても，この私かっこいいんだって。ね，聞いた聞いた？一同：仕事仕事。】

```
					}

					/*---スコアなど表示---*/
					grf.drawString("ＶＥＧＥＴＡＢＬＥ",64,24);
					grf.drawString("ＨＩ－ＳＣＯＲＥ",232,352);
					grf.drawString(""+hisc,224,368);
					grf.drawString("＋＠",272,384);
					grf.drawString(""+scup2,224,400);
					grf.drawString("ＳＣＯＲＥ",256,416);
					grf.drawString(""+score,224,432);
					grf.drawString("ＤＵＢＬＥ",256,448);
					grf.drawString(""+yakudbl,224,464);

					if(gameov){
						grf.setColor(Color.black);
						grf.fillRect(64,128,96,32);
						grf.setColor(Color.white);
						grf.drawString("Take Care",84,148);
					}
					/*---オフスクリーンの絵を実際の画面に表示---*/
					g.drawImage(offs,0,0,this);

					break;

				default:
					break;
		}
	}

	public void start() {		//==================
		kicker = new Thread(this);
		kicker.start();
	}

	public void stop()	{
		kicker = null;
	}

	public void run()	{		//==================

		/*---イメージが読み込まれるまで待つ---*/
		try		{
			mt.waitForID(0);
		} catch (InterruptedException e) {
			return;
		}
		repaint();

		/*---実行中のスレッドをチェック---*/
		Thread thisThread = Thread.currentThread();

		/*---ずっと繰り返し---*/
		while((gameov != true)&&(kicker == thisThread))	{

			ct = ct - 1;
			if(ct<0)	{
				keyflg = false;
```

```
					rakka();
					keyflg = true;
				}

				try		{
					thisThread.sleep(10);
				} catch (InterruptedException e){}
			}
	}

	public void gamestart()		{		//=================
		/*---ゲームスタート---*/
		score = 0;
		scup2 = 0;
		lvr = 1;
		yakudbl = 0;
		gameinit();
		spclv();
		nextvset();
		fallv();
		gameov = false;
		keyflg = true;
		repaint();
		start();
	}

	public void gameover()		{
		/*---ゲームオーバー---*/
		gameov = true;
		keyflg = false;
		stop();
	}

	public void gameinit()		{		//===================
		/*---play画面を何もない状態にする---*/
		for(int i=0 ; i<=vw ; i++)	{
			for(int j=0 ; j<=vh ; j++)	{
				if(i>0 && i<vw && j>0 && j<vh){
					board[i][j] = 0;
				} else {
					board[i][j] = 1 + (int)(Math.random() *
(float)(5));
				}
			}
		}
	}

	public void fallv() {
		/*---次の野菜の落下---*/
		vx = 2;vy = 1;
		board[2][1] = nextvgtbl[0];
		board[3][1] = nextvgtbl[1];
		nextvset();

		rakkasp = 120 - lvr*3;
		if(rakkasp<6)		{
			rakkasp = 6;
```

★大学受かりました。これからは1人暮らしです。もちろん，1人暮らしになってもベーマガは買いつづけますよー！！（愛媛県東宇和郡・毛利祐允18歳）……**【一同：おめでとうございます。影：なんでもセンター試験のBASICはびびって選択しなかったそうです。】**

```
			}
			ct = rakkasp;
	}

	public void nextvset()		{
			/*--次の野菜の決定--*/
			int j = 5;
			if(yakudbl >= 5)	{
					j = 6;
			}
			nextvgtbl[0] = 1 + (int)(Math.random() * (float)(j));
			nextvgtbl[1] = 1 + (int)(Math.random() * (float)(j));
	}

	public void spclv() {
			/*--スペシャルの決定--*/
	  do		{
					spcl = 0;
					for(int i=0 ; i<4 ; i++)		{
							spclvgtbl[i] = 1 + (int)(Math.random() * (float)(5));
							spcl = spcl + keisan[spclvgtbl[i]];
					}
	  } while ((spcl==4) || (spcl==40) || (spcl==400) || (spcl==4000) || (spcl==40000));
	}

	public void rakka() {		//================
			/*--落下処理--*/
			if((board[vx][vy+1] == 0) && (board[vx+1][vy+1] == 0))		{

					/*--下に何もない時--*/
					board[vx][vy+1] = board[vx][vy];
					board[vx+1][vy+1] = board[vx+1][vy];
					board[vx][vy] = 0;
					board[vx+1][vy] = 0;
					vy = vy + 1;
					repaint();
					ct = rakkasp;

			} else {

					/*--下まで落下したか下に何かあるとき--*/
					if((numkey == 0) || (rakkasp-200 > ct)) {
							if(numkey == 0)		{
									keyflg2 = false;
							}
							for(int i=0 ; i<=1 ; i++)		{
									int j = vy;
									while(board[vx+i][j+1] == 0)	{
											board[vx+i][j+1] = board[vx+i][j];
											board[vx+i][j] = 0;
											j++;
									}

									/*--カラスに食べられた--*/
									if(board[vx+i][j] == 6)		{
											board[vx+i][j] = 0;
											if(j<(vh-1))		{
													board[vx+i][j+1] = 0;
											}
									}
							}

							/*--役判定---*/
							scup = 1;spcflg = 0;
							for(x=1 ; x<=5 ; x++)		{
									hantei = 0;
									for(y=(vh-6) ; y<vh ;  y++)		{
											hantei=hantei+keisan[board[x][y]];
											if((hantei==3) || (hantei==30) ||
(hantei==300) || (hantei==3000) || (hantei==30000))
											{
													yaku1(3);
											}
											if(hantei == spcl)	{
													yaku1(4);
													scup = scup *2;
													spcflg = 1;
											}
											if(hantei == 11111)	{
													yaku1(5);
											}
									}
							}

							/*--スコアアップ--*/
							if(scup > 1)		{
									scup2 = scup * lvr;
									score = score + scup2;
									if(score>hisc)		{
											hisc = score;
									}
									lvr = lvr +1;
									if(scup > 5)		{
											yakudbl = yakudbl + 1;
									}
							}

							if(spcflg == 1)		{
									spclv();
							}

							/*--ゲームオーバー判定--*/
							if((board[1][vh-7] + board[2][vh-7] + board[3][vh-7] +
board[4][vh-7] + board[5][vh-7]) > 0)
							{
									gameover();
							} else {
									fallv();
							}

							repaint();
					}
			}
	}

	public void yaku1(int j)		{
```



★突然ですが，影さま。就職活動の極意を教えてください。（神奈川県川崎市・**桜井道孝21歳**）……【**影**：就職活動か，大変だね，がんばって！**つぐ美**：人事のように！影さんを頼ってるのよ。何かアドバイスしてあげたら。**影**：びびらない，あせらない，さわがない。】

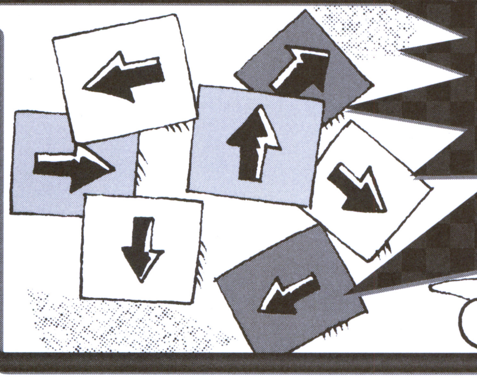
```
            for(int i=(y-(j-1)) ; i<=y ; i++)         {
                    board[x][i] = 0;
            }
            scup = scup * j;
    }

    public void keyTyped(KeyEvent e)         {          //===============
            /*--keyが押されてから離された--*/
    }

    public void keyReleased(KeyEvent e)      {
            /*--keyが離された--*/
            keyflg1 = true;
            keyflg2 = true;
            numkey = -1;
    }

    public void keyPressed(KeyEvent e)       {
            /*--keyが押された--*/
            numkey = e.getKeyCode() - '0' ;          //--数字キーに変換

            if(numkey == 5)     {          //--リプレイ
                    stop();
                    gamestart();
            }

            if(keyflg && (gameov != true)) {

                    if((numkey == 1) && keyflg1)    {          /*--左移動--*/
                            if(board[vx-1][vy] == 0)       {
                                    board[vx-1][vy] = board[vx][vy];
                                    board[vx][vy] = board[vx+1][vy];
                                    board[vx+1][vy] = 0;
                                    vx = vx - 1;
                                    repaint();
                                    keyflg1 = false;
                            }
                    }

                    if((numkey == 2) && keyflg1)    {          /*--右移動--*/
                            if(board[vx+2][vy] == 0)       {
                                    board[vx+2][vy] = board[vx+1][vy];
                                    board[vx+1][vy] = board[vx][vy];
                                    board[vx][vy] = 0;
                                    vx = vx + 1;
                                    repaint();
                                    keyflg1 = false;
                            }
                    }

                    if((numkey == 0) && keyflg2)    {          /*--下移動--*/
                            rakka();
                    }

                    if((numkey == 9) && keyflg1)    {          /*--左右入れ替え--*/
                            int j = board[vx][vy];
                            board[vx][vy] = board[vx+1][vy];
                                    board[vx+1][vy] = j;
                                    repaint();
                                    keyflg1 = false;
                            }
                    }
            }
}
```

### 《リスト２》vegetable3.htmlのプログラム・リスト

```
<HTML>
<HEAD>
<TITLE>vegetable3</TITLE>
<BODY BGCOLOR = "#ffffff">

<table border = 0>
<tr>
<td>

<h1>Vegetable3</h1>
<applet code = "vegetable3.class" width=320 height=512>
<param name = "imagefile.1" value = "img1.gif">
<param name = "imagefile.2" value = "img2.gif">
<param name = "imagefile.3" value = "img3.gif">
<param name = "imagefile.4" value = "img4.gif">
<param name = "imagefile.5" value = "img5.gif">
<param name = "imagefile.6" value = "img6.gif">
</applet>

</td>

<td>
<dl>
        <dt><h2>操作方法</h2>
                <dd><p>0 = 強制落下<br>1 = 左移動<br>2 = 右移動<br>9 = 左右入替<br>5 = ゲームスタート<br>(テンキー不可)
        <dt><h2>遊び方</h2>
                <dd><p>1  :  同じ種類の野菜を3個重ねる。<br>
                2  :  5種類の野菜を重ねる。<br>
                3  :  スペシャルの組み合わせに重ねる。(順不同) <br>
                と、得点になる。    2列同時に並べると高得点。(ダブル)
                <p>ダブルを5回するとカラスが出現するようになる。<br>
                カラスは、重なっている野菜の一番上を食べてくれる。
                <p>灰色の領域まで野菜が積み重なるとゲームオーバー。
</dl>
</td>
</table>

</body>
</html>
```

★突然ですが，編集部に遊びに行ってもいいですか？この春から東京で1人暮らしがはじまります。東京の知り合いはベーマガだけなので…って，編集部は僕を知るわけありませんね。(4月から東京都葛飾区・**初男18歳**)……【**影：読者はみんな友達だ！**】

# iモード対応携帯電話(iアプリ)用



矢印の向きに気をつけろ

# Pair Arrow（ペア アロー）

峠 恒司(47歳)

## はじめに・ルール

iモードJavaで作ったパズルゲームです。

ゲームの内容は，同じ色のコマをペアで消していくというものです(**写真1，2**)。

コマには矢印が描かれており，その矢印の方向に他のコマがないと飛んでいきます。コマが端まで飛んでいくとミスになります。すべての矢印を消すとステージクリアになります(**写真3**がクリア直前画面)。

レベル7まであります。毎回，問題面は違うので，何度でも遊べます(**写真4**)。

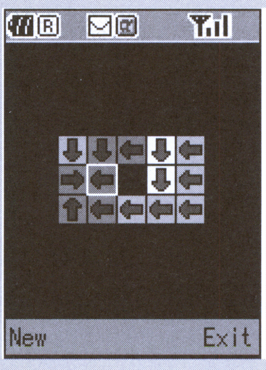
## 操作方法

タイトルが出たら，ソフトキー1(左)を押すとスタート。ゲーム途中で押してもOK。ソフトキー2(右)を押すと終了します。

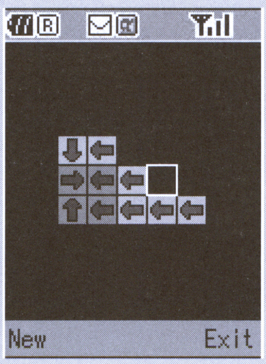
白ワクは↑，←，→，↓キーまたは2，4，6，8キーで動かします。

消したいコマの指定は，セレクトキーまたは5キーを押します。同じ色のコマを二つ指定すると，そのコマを消すことができます。。

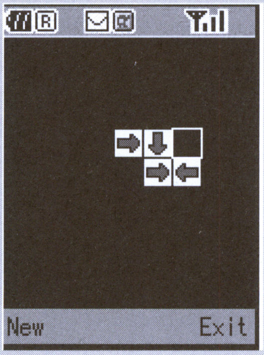
間違って指定したときは，1番目に指定したコマか，別の色のコマの所でセレクトキーを押してください。



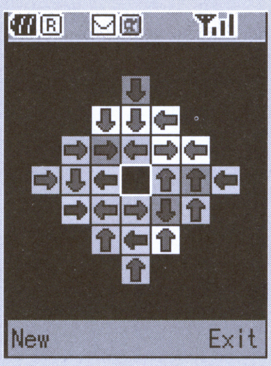

《写真1》矢印を二つ指定して消していくパズル




《写真2》まずは動きをしっかりと理解しよう




《写真3》このような状態になればクリアは目の前




《写真4》先へ進むごとにステージも複雑になっていく

## プログラムについて

特に凝ったところはありません。ミスしたときに振動させようかと思いましたが，バイブレーション機能は機種によって違うようなのであきらめました。

また，タイム表示も最初はしていましたが，どうもキー操作が重たくなるのでこれもやめました。

画像はGIF形式で0.gifは黒一色の空白ブロック，1～16.gifは矢印コマです。矢印の色(正確には背景の色)は，1～4.gifが黄色，5～8.gifが紫色，9～12.gifが黄緑色，13～16.gifが水色となります。

ペイントで描くとファイルサイズが大きくなるので，別の作画ツールで16色に減色してください。

## おわりに

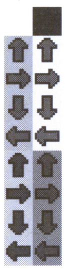
iアプリは2回目の投稿です。携帯は一つしかないので，他の機種でちゃんと動くかどうか不安ではあります。

以前投稿した『ハートをさがせ』(2002年1月号掲載)は，タイマーがエラーになるようです。パソコンのエミュレータではまともに動くのですが。



★大学に合格したら，1人暮らしにあこがれていたんですが，家から通えるということで，親と同居。私の1人暮らしはいつになることやら。(埼玉県上尾市・**大学まで2時間近くかかるんだよー18歳**)……**【つぐ美：それも親孝行のうち。がんばって！】**

入力方法は47ページをご覧ください。



左上から0～16.gifの画像

## CHECKER FLAG

**つぐ美**：ルールの把握にチョット戸惑うかもしれないけど，内容を理解してしまえば，かなり遊べるパズルゲームです。iアプリということもあって待ち合わせ時の時間つぶしなどには最適のゲームだわ。

**編**：オリジナリティをもちつつ，パズルとしての完成度も高い作品です。

**Dr.D**：以前iアプリ制作で気にかけてほしいと言った，方向キー＋数字キーへの対応もしっかり実現されておる。リスト面も，paintメソッド内で言えば，switch文で場合分けして描画処理を行うなど，ていねいな作りがなされておる作品じゃ。また，コメント記述が少ないので，処理内容がチトわかりにくいかもしれんが，ステージ作成部分も非常にこだわった作りと言える。

### 《リスト》Main.javaのプログラム・リスト

```
package pair;                                   //TAORIN SOFT

import java.lang.Math;
import java.util.Random;
import com.nttdocomo.ui.*;

public class Main extends IApplication {
    static MainCanvas mc;
    public void start() {
        mc = new MainCanvas();
        Display.setCurrent(mc);
    }
}

class MainCanvas extends Canvas {
    private int screenW = getWidth();
    private int screenH = getHeight();
    private int dF = 0;                         //フラグ
    private int vF = 2;
    private int aX , aY;                        //白ワク消去座標
    private int bX , bY;                        //白ワク座標
    private int rC , rX , rY;                   //コマ指定
    private int ksu;                            //コマの数
    private int lev = 1;                        //レベル
    private MediaImage mi[] = new MediaImage[17];   //画像読込用
    private Image img[] = new Image[17];
    private int Km[][] = new int[9][9];         //コマの種類
    private int Bm[][] = new int[9][9];

    private String msg = "Pair Arrow";
    static Random rand = new Random();

    MainCanvas() {
        try {
            //画像読込
            for(int imgNo = 16; imgNo >= 0; imgNo--){
                mi[imgNo] = MediaManager.getImage("resource:///"+ imgNo +".gif");
                mi[imgNo].use();
                img[imgNo] = mi[imgNo].getImage();
            }
        }catch (Exception e) {
        }
        setSoftLabel(Frame.SOFT_KEY_1, "New");
        setSoftLabel(Frame.SOFT_KEY_2, "Exit");
        for (int x = 8; x >= 0; x--) {
            for (int y = 8; y >= 0; y--) {
                Km[x][y] = 20; Bm[x][y] = 20;
            }
        }
    }

    //キー操作
    public void processEvent(int type, int param) {
        if (type == Display.KEY_RELEASED_EVENT) {
            switch (param){
            case Display.KEY_SOFT1:
                if (vF != 1) {
                    dF = 9;
                }
                break;
            case Display.KEY_SOFT2:
                IApplication .getCurrentApp().terminate();
                break;
            case Display.KEY_RIGHT: case Display.KEY_6:
                if (vF == 0 && bX < 7) {
                    bX++; dF = 2;
                }
                break;
            case Display.KEY_LEFT: case Display.KEY_4:
                if (vF == 0 && bX > 1) {
                    bX--; dF = 2;
                }
                break;
            case Display.KEY_DOWN: case Display.KEY_8:
                if (vF == 0 && bY < 7) {
                    bY++; dF = 2;
                }
                break;
            case Display.KEY_UP: case Display.KEY_2:
                if (vF == 0 && bY > 1) {
                    bY--; dF = 2;
                }
                break;
            case Display.KEY_SELECT: case Display.KEY_5:
                if (vF == 0 && Km[bX][bY] > 0){
                    if (rC == 0) {
                        rC = (Km[bX][bY] - 1) / 4 + 1;
                        rX = bX; rY = bY;
                        dF = 3;
                    } else if((rX == bX && rY == bY) || rC != (Km[bX][bY] - 1) / 4 + 1){
```

★自動車学校通い始めたんですが，いやはや「リッジレーサー」で世界をとった私もタジタジでござるよ。（東京都中央区・**まぢけん20歳**）……【**編**：影さんにハンドル握らせるとゲーム感覚で突っ走るから怖いんですけど。**つぐ美**：自転車の2人乗りはいけません。】

```
                        dF = 5;
                    } else {
                        dF = 4;
                    }
                }
                break;
            }
            repaint();
        }
    }

    public void paint(Graphics g) {
        int f, j, x , y;
        int gX = (screenW - 98) / 2;
        int gY = (screenH - 98) / 2;
        switch (dF){
            case 0:
                //タイトル表示
                g.lock();
                g.setColor(g.getColorOfName(g.YELLOW));
                g.fillRect(0, 0, getWidth(), getHeight());
                g.setColor(g.getColorOfName(g.BLACK));
                g.drawString(msg, (screenW - 60) / 2 + 1, 40);
                g.setColor(g.getColorOfName(g.RED));
                g.drawString(msg, (screenW - 60) / 2, 40);
                g.drawImage(img[7], gX + 35, 50);
                g.drawImage(img[9], gX + 49, 50);
                g.unlock(true);
                break;
            case 1:
                //面表示
                g.lock();
                g.setColor(g.getColorOfName(g.BLACK));
                g.fillRect(0, 0, getWidth(), getHeight());
                g.setColor(g.getColorOfName(g.BLACK));
                g.fillRect(gX - 1, gY - 1, 99, 99);
                for (x = 7; x > 0; x--) {
                    for (y = 7; y > 0; y--) {
                        g.drawImage(img[Km[x][y]], gX + 14 * (x - 1), gY + 14 * (y - 1));
                    }
                }
                g.unlock(true);
                aX = 4; aY = 4;
                bX = 4; bY = 4;
                rC = 0; rX = 0; rY = 0;
                dF = 2;
                repaint();
                break;
            case 2:
                //白ワク表示
                vF = 1;
                g.lock();
                g.setColor(g.getColorOfName(g.BLACK));
                g.drawRect(gX + 14 * aX - 15, gY + 14 * aY - 15, 14, 14);
                g.setColor(g.getColorOfName(g.WHITE));
                g.drawRect(gX + 14 * bX - 15, gY + 14 * bY - 15, 14, 14);
                g.unlock(true);
```

```
                aX = bX; aY = bY;
                dF = 10; vF = 0;
                break;
            case 3:
                //１番目のコマの指定
                g.lock();
                g.setColor(g.getColorOfName(g.RED));
                g.drawRect(gX + 14 * bX - 14, gY + 14 * bY - 14, 12, 12);
                g.drawRect(gX + 14 * bX - 11, gY + 14 * bY - 11, 6, 6);
                g.unlock(true);
                dF = 10; vF = 0;
                break;
            case 4:
                //２つのコマ消す
                g.lock();
                Km[bX][bY] = 0; Km[rX][rY] = 0;
                g.drawImage(img[0], gX + 14 * (bX - 1), gY + 14 * (bY - 1));
                g.drawImage(img[0], gX + 14 * (rX - 1), gY + 14 * (rY - 1));
                g.unlock(true);
                rC = 0; rX = 0; rY = 0;
                dF = 6;
                repaint();
                break;
            case 5:
                //キャンセル
                g.lock();
                g.drawImage(img[Km[rX][rY]], gX + 14 * (rX - 1), gY + 14 * (rY - 1));
                g.unlock(true);
                rC = 0; rX = 0; rY = 0;
                dF = 10; vF = 0;
                break;
            case 6:
                //コマ移動
                int x1 , y1;
                g.lock();
                g.setColor(g.getColorOfName(g.RED));
                f = 0;
                for (y = 1; y < 8; y++) {
                    for (x = 1; x < 8; x++) {
                        x1 = 0; y1 = 0;
                        switch (Km[x][y]){
                            case 1: case 5: case 9: case 13:
                                x1 = -1; y1 = 0;
                                break;
                            case 3: case 7: case 11: case 15:
                                x1 = 1; y1 = 0;
                                break;
                            case 2: case 6: case 10: case 14:
                                x1 = 0; y1 = -1;
                                break;
                            case 4: case 8: case 12: case 16:
                                x1 = 0; y1 = 1;
                                break;
                        }
                        if (x1 + y1 != 0) {
                            if (Km[x + x1][y + y1] == 0) {
                                f = 1;
```



★4月号87ページのカツラほしい！／友人が合格祝いにVAIOを買ってもらったそうです。／ベーマガ発売日に風邪で寝込んでしまうあたり僕は未熟者です。／衝撃の事実！つぐ美さんは「つぐよし」さんだった。（東京都練馬区・**ルストフレム15歳**）……**【影：はっ！】**

```
                g.drawImage(img[0],gX+14*(x-1),gY+14*(y-1));
                g.drawImage(img[Km[x][y]],gX+14*(x+x1-1),gY+14*(y+y1-1));
                Km[x + x1][y + y1] = Km[x][y]; Km[x][y] = 0;
            } else if (Km[x + x1][y + y1] == 20) {
                dF = 8;
                g.drawLine(gX+14*(x-1),gY+14*(y-1),gX+14*x-2,gY+14*y-2);
                g.drawLine(gX+14*(x-1),gY+14*y-2,gX+14*x-2,gY+14*(y-1));
            }
        }
    }
}
g.unlock(true);
if (f == 1 && dF == 8) {
    dF = 6;
} else if (f == 0 && dF == 6) {
    dF = 10; vF = 0;
    //クリア?
    ksu —;
    if (ksu < 1){
        dF = 7; vF = 1;
    }
}
repaint();
break;
case 7:
    msg = "LEVEL" + lev +" クリア";
    g.lock();
    g.setColor(g.getColorOfName(g.AQUA));
    g.fillRect(0, 0, getWidth(), getHeight());
    g.setColor(g.getColorOfName(g.BLACK));
    g.drawString(msg, (screenW - 72) / 2 + 1, screenH / 2 + 4);
    g.setColor(g.getColorOfName(g.BLUE));
    g.drawString(msg, (screenW - 72) / 2 , screenH / 2 + 3);
    g.unlock(true);
    lev ++;
    if (lev > 5){
        lev = 5;
    }
    vF = 2; dF = 10;
    break;
case 8:
    msg = "ﾐｽ!! ﾐｽ!! ﾐｽ!!";
    g.lock();
    g.setColor(g.getColorOfName(g.GRAY));
    g.drawString(msg, (screenW - 80) / 2 + 1, screenH / 2 + 4);
    g.setColor(g.getColorOfName(g.WHITE));
    g.drawString(msg, (screenW - 80) / 2 , screenH / 2 + 3);
    g.unlock(true);
    vF = 2; dF = 10;
    break;
case 9:
    msg = "面作成中...";
    g.lock();
    g.setColor(g.getColorOfName(g.BLACK));
    g.fillRect(0, 0, getWidth(), getHeight());
    g.setColor(g.getColorOfName(g.WHITE));
    g.drawString(msg, (screenW - 60) / 2 , screenH / 2 + 3);
```

```
            g.unlock(true);
            newGame();
            dF = 1;
            repaint();
            break;
        }
    }
    //面作成
    private void newGame(){
        int f, g, h, i, j, t, x, y;
        int x2 = 0, y2 = 0;
        for (x = 7; x > 0; x—) {
            for (y = 7; y > 0; y—) {
                Km[x][y] = 0; Bm[x][y] = 20;
            }
        }
        switch (lev){
            case 1:
                ksu = 7;
                for (x = 6; x > 1; x--) {
                    for (y = 5; y > 2; y--) {
                        Bm[x][y] = 0;
                    }
                }
                Bm[4][4] = 20;
                break;
            case 2:
                ksu = 8;
                for (x = 2; x < 5; x++) {
                    for (y = 2; y < 5; y++) {
                        Bm[x][y] = 0;
                    }
                }
                for (x = 4; x < 7; x++) {
                    for (y = 4; y < 7; y++) {
                        Bm[x][y] = 0;
                    }
                }
                break;
            case 3:
                ksu = 12;
                for (x = 6; x > 1; x--) {
                    for (y = 6; y > 1; y--) {
                        Bm[x][y] = 0;
                    }
                }
                Bm[4][1] = 0; Bm[4][7] = 0;
                Bm[1][4] = 0; Bm[7][4] = 0;
                Bm[2][2] = 20; Bm[6][2] = 20;
                Bm[2][6] = 20; Bm[6][6] = 20;
                Bm[4][4] = 20;
                break;
            case 4:
                ksu = 17;
                for (x = 7; x > 0; x--) {
                    for (y = 7; y > 0; y--) {
                        Bm[x][y] = 0;
```



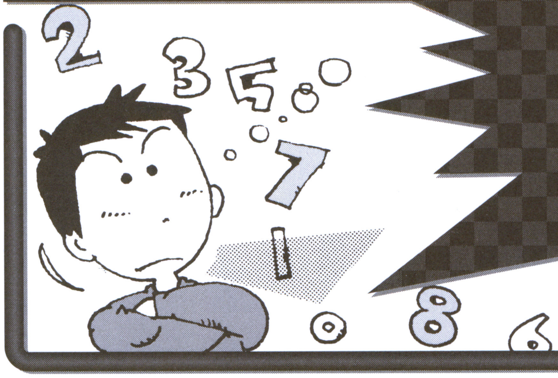
★カレンダー当たりました。ありがとうございました。影さん，直接届けに来るなら一報入れといてくださいよ。パソコンが壊れてましたよ，マジで（涙）。（大阪府門真市・**上7 28歳**）……**【影：失礼！パソコンが留守番していたもんだから，ついつい長話しちゃって。】**

```
                        }
                }
                for (x = 5; x > 0; x--) {
                    for (y = 5; y > 2; y--) {
                        Bm[x][y] = 20;
                    }
                }
                break;
            case 5:
                ksu = 20;
                for (x = 7; x > 0; x--) {
                    for (y = 7; y > 0; y--) {
                        Bm[x][y] = 0;
                    }
                }
                for (x = 5; x > 2; x--) {
                    for (y = 5; y > 2; y--) {
                        Bm[x][y] = 20;
                    }
                }
                break;
            case 6:
                ksu = 22;
                for (x = 7; x > 0; x--) {
                    for (y = 7; y > 0; y--) {
                        if (x != 4){
                            Bm[x][y] = 0;
                        }
                    }
                }
                Bm[4][3] = 0; Bm[4][5] = 0;
                break;
            case 7:
                ksu = 24;
                for (x = 7; x > 0; x--) {
                    for (y = 7; y > 0; y--) {
                        Bm[x][y] = 0;
                    }
                }
                Bm[4][4] = 20;
                break;
        }
        rC = Math.abs(rand.nextInt() % 4);
        do{
            x = Math.abs(rand.nextInt() % 7) + 1;
            y = Math.abs(rand.nextInt() % 7) + 1;
        } while (Bm[x][y] != 0 || Bm[x + 1][y] != 0);
        Bm[x][y] = 3 + 4 * rC; Bm[x + 1][y] = 1 + 4 * rC;
        for (j = ksu - 1; j > 0; j--) {
            rC = Math.abs(rand.nextInt() % 4);
            g = 0;
            do{
                for (i = 1; i >= 0; i--) {
                    f = 0;
                    do{
                        h = 0;
                        do{
                            x = Math.abs(rand.nextInt() % 7) + 1;
                            y = Math.abs(rand.nextInt() % 7) + 1;
                        }while (Bm[x][y] == 20);
                        if (Bm[x][y] > 0){
                            switch (Bm[x][y]){
                                case 1: case 5: case 9: case 13:
                                    x2 = 1; y2 = 0; break;
                                case 3: case 7: case 11: case 15:
                                    x2 = -1; y2 = 0; break;
                                case 2: case 6: case 10: case 14:
                                    x2 = 0; y2 = 1; break;
                                case 4: case 8: case 12: case 16:
                                    x2 = 0; y2 = -1; break;
                            }
                            if (Bm[x + x2][y + y2] == 0){
                                h = 1;
                                Bm[x + x2][y + y2] = Bm[x][y]; Bm[x][y] = 0;
                            }
                        }
                        if (Bm[x][y] == 0){
                            do {
                                t = Math.abs(rand.nextInt() % 4) + 1;
                                switch (t){
                                    case 1:
                                        x2 = -1; y2 = 0; break;
                                    case 2:
                                        x2 = 0; y2 = -1; break;
                                    case 3:
                                        x2 = 1; y2 = 0; break;
                                    case 4:
                                        x2 = 0; y2 = 1; break;
                                }
                                if (Bm[x + x2][y + y2] > 0 && Bm[x + x2][y + y2] < 20){
                                    f = 1; Bm[x][y] = t + 4 * rC;
                                }
                            } while (h == 1 && f == 0);
                        }
                    } while (f == 0);
                }
                g = 1; break;
            }while (g == 0);
        }
        for (x = 7; x > 0; x--) {
            for (y = 7; y > 0; y--) {
                if (Bm[x][y] < 20){
                    Km[x][y] = Bm[x][y];
                }
            }
        }
        dF = 1;
    }
}
```



★「見やすいサイトね」「人と同じく見ばえが大事か…」「フフ…外見だけじゃ…」（愛知県名古屋市・ズオウ・リアキ29歳）……**【影：だから誰と誰が会話してるんだよ！つぐ美：誰は関係ないと思うんですが？1人芝居かもよ。影：♪チャンチャラチャラチャンチャン。】**

# PIECE用




PIECE投稿作品第１弾

# Nazo（ナゾ）

谷澤　友哉（14歳）

## 内容

このプログラムは，PIECEで動くゲームです。僕的には今，PIECEのソフトを投稿すれば載るのではないか，と思ったから投稿しました。過去にもiアプリを初掲載ということで載りましたので,今回も…と考えたのが，この数探りゲーム『Nazo』です(ちなみに，iアプリについては2001年７月号に載りました～！)。

ということはさておき，プログラムの内容についてですが，これは数探りゲームというものです(自分で勝手に言っているのですが…)。

このゲームでは，指定された最高値までプレイヤーとコンピュータが順々に値を加えていき，最高値を超してしまったほうが負け，という単純なゲームです。ですが，このゲーム，単純なんですがコンピュータに勝つのは少々難しいです。けっこう頭を使うゲームなんですよ,これは。とにかく，コンピュータの頭脳に勝ってくださいね！

## 使い方・遊び方

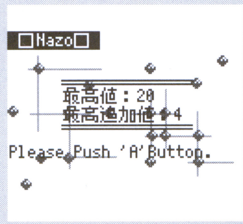
『Nazo』の遊び方は,いたって簡単です。とにかくカンタンです。あとは頭を使うだけです(笑)。

では，説明をします。この数探りゲームというゲームはとにかく，コンピュータと競い，相手に最高値をとらせるというゲームです。この手のゲームは，１回はどこかでやったことがあるのではないかと思いますが，一応説明しときましょう。

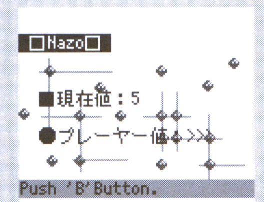
こっちは相手に最高値を取らせるべく，計算して追加値を入力していけばいいのです。もちろんコンピュータも追加値を入力してくるので，そのことも考えて数を入力していってください。

ここからは，実際に追加値の入力方法を説明します。追加値の入力方法は，「プレーヤー：>」と追加値入力プロンプトが出ているときに，左右の方向キーで入力します。追加値は，１～４まで選択可能です。自分の思うとおりの値になったら，Bボタンで決定します。こっちが値を入力すると，すかさずコンピュータが入力してきます。この時にしっかりとコンピュータの入力した値を見ておくことが大切です。これを参考に今後の作戦を立てていくのです。

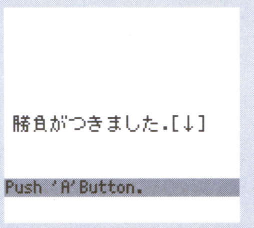
これだけ学習すれば，あとはもうプレイです！。



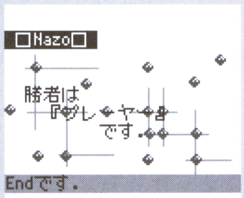
《写真１》本誌初のPIECE投稿プログラム



《写真２》手軽に遊べる数探りゲーム



《写真３》加算した値が20を超えると決着する



《写真４》先を読んだプレイでCOMに勝利しよう1

## プログラムの仕組み

このプログラムは，まだ僕がPIECEによるプログラミングの初心者で，まだあまり多くのPIECE用ソースを見ていないので，その仕組みが,まだあいまいですが,一応説明をしておきます。

ゲーム処理自体は，pceAppProc内ですべて行っています。すべてを行う方法としてcScene変数を利用してゲームの場面を管理しています。cScene変数を利用するシーン管理プログラミング以外は，まだP/ECEプログラミングでは行ったことがないので,どうすればいいかわかりません。Windowsプログラミング(C++)も



入力方法は50ページをご覧ください。

微妙なので…(笑)。

## プログラムの入力方法

main.c(**リスト1**)と，sub.c(**リスト2**)と，nazo.h(**リスト3**)をそれぞれ入力し，保存します。そして，back.c(画像データ。back.bmpからPIECE付属のツールで作成)と，PIECEに付属しているwavetbl.cならびに，PIECEチュートリアルで使用するsample.mmlをbackmusic.mmlとリネームしたファイルを混ぜ，さらにコマンドラインで「pcc33 main.c sub.c back.c wavetbl.c」としてコンパイルをしてください。

その後，runコマンド，もしくはPIECE本体へ転送して，PIECE本体で実行してみてください。

## 改造

このプログラムは，改造しやすいように最高値と追加値を変更できるようになっています。方法はnazo.h内のVALUE_MAX，VALUE_MOVEの定数定義を変更するだけです。こんだけだったら改造とは言えません(たぶん)。この改造の本番はここからです。

改造する対象は最高値，追加値にあわせ，コンピュータの思考ルーチンを再記述することです。コンピュータの思考ルーチンはsub.c(**リスト2**)内のGetNicePoint()関数で定義されていますので，どうぞよりよい改造を。

## 作者からのコメント

PIECEでのプログラミングは楽しいです。前にやったｉアプリも楽しかったですが，これは実機がなかったので最後の最後であきらめてしまいました。が，PIECEは9800円という価格でCコンパイラ付き！　買ったらすぐプログラミングができます(できました)。とにかく，楽し〜い。

## 宣伝

この『Nazo』とは関係なくなってしまうんですが，本誌に投稿する際に強い味方になってくれるだろうものを紹介しておきます。

というのは投稿する際，アイデアとして味気なくなったとき，アイデアが浮かばないとき，アイデアの参考を見たいとき，ソフトのアイデアが豊富にあるHPがあるんです(やっぱり宣伝です)。それはアイディアFactoryというものです。ここでみつけたアイデアで一発投稿してみてはいかがでしょうか。

**アイディアFactory**＝http://hp.vector.co.jp/authors/VA025701/FrameIdeaFactory.html

また，このアイディアFctoryも底なしというわけにはいきませんから，どうか，アイデア投稿もお願いしますネ。詳細は上記URLを。

## 特別攻略法

この項目はゲームをどうしてもクリアできない場合読んでください。

この数探りゲームというものは，やっぱり一定のルールの上に成り立っているゲームなので，やはり法則があり，クリアするための手段というものがあります。その手段というものは，こちらで19をカウントさせるというものです。単純です。こうすればコンピュータは必然的に20をカウントすることになります。こちらで19をカウントさせるためには，こちらが16をカウントできていればいいということです。つまり，コンピュータに14をカウントさせるのです。

どうでしょうか，ここまで言えばクリアできるでしょうか。まだまだ法則にのっとった手段はありますが，ここでは伏せておきましょうか…。自分で発見してみてください。ということです(笑)。

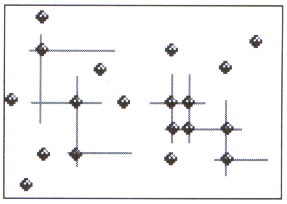
### CHECKER FLAG

**編**：谷澤さんの狙いどおり，PIECEでの掲載第１弾となりました。PIECE本体の性能は，パソコンなどと比べれば劣るものですが，その分プログラミングという面では，制作者の実力が発揮できるのではないでしょうか？ぜひとも本誌の読者にもPIECEプログラミングに挑戦してほしいですね。

**つぐ美**：コンピュータに勝つのは難しいと書いてありますが，特別攻略法に書いてある法則に気付いてしまえば簡単に勝てます。この手のゲームでは基本と言える手法ですね。

**Dr.D**：指摘した手法をやりにくくするため，指定できる値の数を制限するなどの工夫をすれば，よりゲーム性が増すじゃろう。

**影**：たまにコンピュータが思考したまま止まってしまうというバグがあるのが残念だな。



**BACK.bmpの画像**

《リスト１》main.cのプログラム・リスト

```
//*** main.c ****************
#include <string.h>
#include <piece.h>
#include "nazo.h"

unsigned char vbuff[128*88];
extern unsigned char BACK[];
int cScene=0;
int movePoint=1;
int gamePoint=0;
int flg=0;

DRAW_OBJECT draw_obj;
PIECE_BMP back_bmp;
```

★つぐ美さんのことを「つぐ美ちゃん」と言わない僕の1/3の純情。(福岡市・**前田宣弘28歳**)……【**編**：影さんも幼少の頃は「影ちゃん」と呼ばれてたんですか？**影**：近所のおばさんからは「影ちゃん，影ちゃん」とかわいがられていて，いつも叱られていた。**編**：は？】

```c
unsigned char snd[] = {
#include "backmusic.c"
};

void pceAppInit( void )
 {
        pceLCDDispStop();
        pceLCDSetBuffer( vbuff );
        pceAppSetProcPeriod( 10 );
        memset( vbuff, 0, 128*88 );

        GameScreen();

        pceLCDLine(3, (128/2)-(70/2), 28, (128/2)+(70/2), 28);
        pceLCDLine(3, (128/2)-(70/2), 26, (128/2)+(70/2), 26);
        pceFontSetTxColor(3);
        pceFontSetPos((128/2)-(70/2), 30);
        pceFontPrintf("最高値：%d", VALUE_MAX);
        pceFontSetTxColor(3);
        pceFontSetPos((128/2)-(70/2), 40);
        pceFontPrintf("最高追加値：%d", VALUE_MOVE);
        pceLCDLine(3, (128/2)-(70/2), 50, (128/2)+(70/2), 50);
        pceLCDLine(3, (128/2)-(70/2), 52, (128/2)+(70/2), 52);
        pceFontSetTxColor(3);
        pceFontSetPos(0, 60);
        pceFontPrintf("Please Push 'A' Button.");
        pceLCDTrans();

        InitMusic();
        PlayMusic(snd);
        pceLCDDispStart();
 }

void pceAppProc( int cnt )
 {
        int pad=pcePadGet();
        int count;
        PCETIME s_time, e_time;

        switch (cScene) {
                case 0:
                //最初のタイトル画面の 時の処理
                        if (pad&TRG_A) {
                                cScene = 1;
                                GameScreen();
                         }
                        break;
                case 1:
                //ゲーム_プレーヤー入力
                        MSG("Push 'B' Button.");
                        pceFontSetPos(10, 50);
                        pceFontPrintf("●プレーヤー値：>");
                        pceLCDTrans();
                        if (pad&TRG_LF) movePoint--;
                        else if (pad&TRG_RI) movePoint++;
                        else if (pad&TRG_B) {
                                //入力値を決定
                                flg = 3;
                                cScene = 2;
                                break;
                        }
                        else break;

                        if (movePoint<1) movePoint=1;
                        else if (movePoint>VALUE_MOVE) movePoint=VALUE_MOVE;
                        GameScreen();
                        MSG("Push 'B' Button.");
                        pceFontSetPos(10, 50);
                        pceFontPrintf("●プレーヤー値：");
                        pceLCDTrans();
                        ShowMove();
                        break;
                case 2:
                //値を参照
                        gamePoint += movePoint;
                        GameScreen();
                        if (gamePoint>=VALUE_MAX) {
                                //ゲーム結果が出たときの処理
                                gamePoint=VALUE_MAX;
                                cScene = 10;
                                break;
                        }
                        if (flg==1) movePoint = 1;
                        cScene = flg;
                        break;
                case 3:
                //ゲーム_相手プレーヤー入力
                        pceFontSetPos(10, 60);
                        pceFontPrintf("○コンピューター：>");
                        pceLCDTrans(); MSG("Push 'B'Button.");

                        movePoint = GetNicePoint();
                        if (movePoint<1) movePoint=1;
                        else if (movePoint>VALUE_MOVE) movePoint=VALUE_MOVE;
                        GameScreen();
                        MSG("Computer thinks.");
                        pceFontSetPos(10, 60);
                        pceFontPrintf("○コンピューター：");
                        pceLCDTrans();
                        ShowMove();
                        for (pceTimeGet(&s_time), pceTimeGet(&e_time); e_time.ss-
s_time.ss!=1; pceTimeGet(&e_time));
                        flg = 1; cScene = 2;
                        break;
                case 4:
                //結果を表示
                        cScene = 0; GameScreen();
                        pceFontSetPos(10, 30); pceFontPrintf("勝者は");
                        pceFontSetPos(20, 40);
                        if (flg==1) pceFontPrintf("『プレーヤー』");
                        else pceFontPrintf("『コンピューター』");
                        pceFontSetPos(50, 50); pceFontPrintf("です.");
```

PIECE

```
                    pceLCDTrans(); cScene = 5; MSG("Endです.");
                    if (pad&TRG_B) cScene = 5;
                    break;
            case 5:
            //最後の待機
                    break;
            case 10:
            //結果処理
                    pceLCDPaint(0, 0, 0, 128, 88);
                    pceFontSetPos(128/2 - 120/2, 88/2);
                    pceFontPrintf("勝負がつきました.[↓]");
                    pceLCDTrans(); MSG("Push 'A'Button.");
                    if (pad&TRG_A) cScene = 4;
                    break;
        }
}

void pceAppExit( void )
{
        pceWaveStop(0);
}
```

## 《リスト2》sub.cのプログラム・リスト

```
//*** main.c ****************
#include <string.h>
#include <piece.h>
#include "nazo.h"

extern unsigned char vbuff[128*88];
extern unsigned char BACK[];
extern int cScene;
extern int movePoint;
extern int gamePoint;
extern DRAW_OBJECT draw_obj;
extern PIECE_BMP back_bmp;

void MSG(char *str)
{
        pceLCDPaint(1, 0, 78, 128, 88);
        pceFontSetPos(0, 78);
        pceFontPrintf("%s", str);
        pceLCDTrans();
}

void GameScreen(void)
{
        pceLCDDispStop();
        pceLCDPaint(0, 0, 0, 128, 88);

        memcpy(&back_bmp.header, BACK, sizeof(PBMP_FILEHEADER));
        back_bmp.buf = BACK+sizeof(PBMP_FILEHEADER);
        pceLCDSetObject(&draw_obj, &back_bmp, 0, 0, 0, 0, 128, 88, DRW_NOMAL);
        pceLCDDrawObject(draw_obj);
        pceFontSetBkColor(3); pceFontSetTxColor(0); pceFontSetPos(0, 0);
        pceFontPrintf("%s", APP_TITLE);
        pceFontSetBkColor(-1); pceFontSetTxColor(3);

        if (cScene) {
                pceFontSetTxColor(3);
                pceFontSetPos(10, 30);
                pceFontPrintf("■現在値:%d", gamePoint);
        }
        pceLCDTrans(); pceLCDDispStart();
}

void ShowMove(void)
{
        int count;

        pceLCDDispStop();
        for (count=1; count<=movePoint; count++) {
                pceFontPrintf(">");
        }
        pceLCDTrans(); pceLCDDispStart();
}

int GetNicePoint(void)
{
        int value;

        if (gamePoint>=16) {
                //勝利適用
                return 19-gamePoint;
        }
        else if(gamePoint>=6 && gamePoint<9) {
                //有利手適用
                return 9-gamePoint;
        }
        else {
                srand(movePoint);
                value = rand()%(VALUE_MOVE+1);
                return value;
        }
}
```

## 《リスト3》nazo.hのプログラム・リスト

```
//*** nazo.h ****************

//定数定義
#define APP_VER          "1.0"
#define APP_TITLE      " □Nazo□ "
#define VALUE_MAX    20
#define VALUE_MOVE   4
//プロトタイプ宣言
void MSG(char *str);
void ShowMove(void);
void GameScreen(void);
int GetNicePoint(void);
```

★Windows XPを買いました。ハードディスク，CPUが巨大な大きさなのに，この遅い動きは何だ！！！ふざけんな！いったい何のための最新機種なんだ。おそい。おもい。（静岡市・**コージイグ20歳**）……**【編集長**：うーん，私もXPなんだけど，快適に動いてるぞ。**】**

★★★★★★★ マイコンBASICマガジン ★★★★★★★

# 明日のスター・プログラマー

## Programmer's Room

### 今月の投稿をズバリ解析！

**編：**今月は，投稿作品では初掲載となるPIECEを含め，さまざまな言語での作品が採用されました。こういった多種多様な言語の作品を一度に掲載できるのも，ベーマガならではのテイストと言えるかもしれませんね。そんな投稿作品の中から，今月もベストプログラムを選考するとしましょう。

**つぐ美：**チェックしていて面白かった作品は，HSP用の『Defense-line』と，iアプリ用の『Pair Arrow』ですね。両作品とも，ゲームとしての面白味をストレートに表現している作品じゃないかしら？

**影：**超大作であるVisual Basic+DirectXの2作品も，3D作品ということで目についたけど，ゲーム自体の内容面でもうひとつ面白味に欠ける印象だった。面白かった作品といえば，やはり『Pair Arrow』かな？手軽に遊べる内容なので，iアプリで遊ぶにはもってこいの作品だね。

**Dr.D:** ワシとしては，初掲載となるPIECE用の『Nazo』に注目したいところじゃが，チトゲーム内容が単純なのが残念なところじゃな。個人的にもPIECEプログラムは，もっと盛り上がっていってほしいと思っておるので，この作品をキッカケに，PIECEの投稿作品が増えてほしいものじゃな。ベストに選ぶとしたら，皆がすでに述べている『Pair Arrow』がふさわしいのではないかな？ゲーム内容もしっかりしているが，リスト面のレベルも高い作品だと思うゾ。

**編：**今月は満場一致といった感じですね。それでは，今月のベスト・プログラマーは，iアプリ用『Pair Arrow』の，峠　恒司さんに決定です。受賞おめでとうございます。

### 投稿する際の注意事項

**編：**ここ最近は，投稿の心得(2月号掲載)や，採用へのウラ技(3月号掲載)などを取り上げてきましたが，その投稿に関する注意事項については，あまり触れていませんでしたね。投稿に関しては，読者から質問が寄せられていたり，また最近になってチョット不備が目立つようになってきたので，ここで一度注意事項をまとめておきましょう。

#### 郵送・メール投稿共通

・ペンネーム希望者の方は，ペンネームをわかりやすい場所(できれば本名を記載している場所すべて)に記述してください。

・作品タイトルを記述していない投稿が，最近目につきます。意外と多い不備の一つですので，投稿前にしっかりとチェックしてください。

#### 郵送投稿の注意点

・作品を郵送する際には，普通郵便で大丈夫です。たまに書留や速達で郵送される方もいらっしゃるのですが，普通郵便で問題なく到着しますのでご安心を。

・プログラムのリストを印刷して同封してくださる方もいらっしゃるのですが，印刷したリストを同封していただく必要はありません。

#### メール投稿の注意点

・「メール本文に何か書くの？」という質問をいただくのですが，メール本文には，基本的に何も記述しなくてOKです。たまに，メール本文の部分に作品の説明文を記述される方がいらっしゃるのですが，これは避けてください。説明文は，添付していただく作品の圧縮ファイル内に，説明文を記述したテキストファイルを同封してください。

・たまに，「このHP上にある作品を見てください」といった感じで，HPのURLが書かれた投稿もあるのですが，こういった形式の投稿は受け付けていませんので，ご注意ください。プログラム投稿は，本誌やベーマガHP上に記載されている形式に沿った形でお願いします。

### 今月はもう一歩でした

**●QBasic用『剰余芸術』／石山明弘(33歳)**

**編：**数学的な式を利用して，模様を描き出すというソフト。着眼点は面白く，内容もなかなか興味ぶいものでしたが，やや単調な印象もあります。実用面なども含めて，もう一歩こだわりがほしいところでした。

**●VisualBasic用『チキチキバンバン3』／梶原寿樹(29歳)**

**影：**3月号掲載作品の続編にあたる作品。オリジナルなアイデアが追加されてはいたけど，もう一つ新鮮味がないという印象だった。

**●VisualBasic用『Light+』／清家　巧(17歳)**

**つぐ美：**上の作品と同じく3月号掲載作品の改良版と言える作品ですね。CHECKER FLAGで指摘したような改良は施されていたのですが，やはり前作と大きな違いがないため，新鮮味がなかったですね。

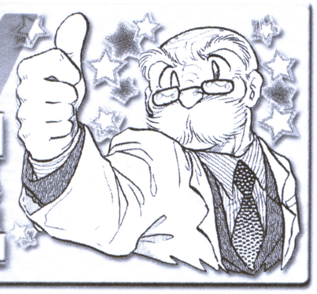
**●PC9801 N88-BASIC用『WinBit』／Y田@(27歳)**

**Dr.D：**N88-BASICの画面をBMPファイルへ保存してくれる実用ソフト。技術面では，なかなか面白味があったのじゃが，実用面や使い勝手の面で，あと一歩及ばなかったのぉ。

### 投稿ありがとう

福崎忠彦／星　裕一郎／有田安彦／内田　寛／峠　恒司／菅野友紀／Y田@／谷澤友哉／伊藤隆朗／橘　一馬／長谷川　誠／MONOMON／清家　巧／米久保　直樹／梶原寿樹／金井義男／くろすけ／石山明弘／
※全員紹介できなくてごめんなさい。

# Dr.Dのベスト・プログラム研究室



## 今月のベスト・プログラム：iアプリ用『Pair Arrow』(88ページ掲載)

今月のベストプログラムは，iアプリ用『Pair Arrow』。パズルゲームとしてのアイデアや完成度，細かな配慮がなされたリストが評価された作品じゃ。今回は，ステージ作成処理の解説と，行った手順を一つ前に戻すアンドゥ機能を付ける改造を紹介していこう。

(Dr.D)

### ステージ作成処理

まず，リスト面での解説を行う前に，このようなパズルゲームのステージが，プログラム内で，どのように実現されているかを簡単に説明しよう。

今回のような形式のパズルゲームでは，ほとんどの作品が，配列を用いた仮想画面による処理を行っている。仮想画面処理というのは，ステージ上の物体(ブロックや壁)をすべて数値で示し，その数値をステージを示す配列へ代入し，配列の位置と値から，ステージ上のどこにブロックや壁があるかを調べる，というものじゃ。今回の作品も，ステージデータの保存には，Km[][]という2次元配列を用いておる。それでは，ステージの基本的な処理を学んだところで，このプログラムでの処理を解説していこう。

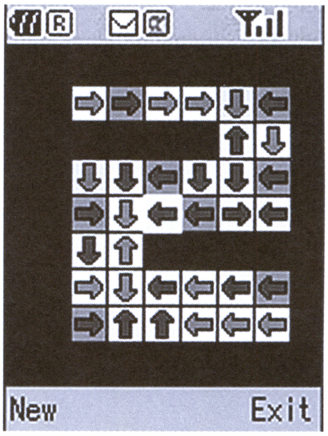
リスト内でステージを作成しているのは，newGame()というメソッド(関数)になる。このメソッド内の処理を一つ一つ追って解説していくとしよう。

まず最初に行うのが，for文を利用しての配列の初期化じゃな。ここは，配列に値を代入しているだけなので，特に問題ない処理じゃろう。

次に行われているのが，switch文での各ステージごとの処理。ここの処理では，ブロックを配置するステージの形を定義しておるのじゃ。処理内容としては，配列Bmに対して0を代入したところにはブロックを置き，20を代入したところは空白にする，という処理になる。試しに，この部分を**リスト1**のように変更して実行してみると，**写真1**のような形のステージを定義することができる。

《リスト1》ステージの形を変更するリスト

```
case 1:
        //ksu×2の値が，配置されるブロック数になる。
        //ステージを自作する際には，この値と，ステージ上に
        //配置するブロック数を，同じにする必要がある。
ksu = 17;
for (x = 7; x > 1; x--) {
        for (y = 7; y > 0; y--) {
    Bm[x][y] = 0; //この場所にブロックを配置
        }
}
        //ブロックを置かない場所を定義(空白になる)
        for(x=0; x<6; x++)
        Bm[x][2] = 20;
        for(x=4; x<8; x++)
        Bm[x][5] = 20;
break;
```




《写真1》改造ステージ例

ステージの形を定義したあとは，いよいよ矢印ブロックの配置となるのじゃが，このプログラムでは，いきなり矢印ブロックをランダムに配置していくのではなく，まず最初に→←ペアのブロックを配置している。これは，いわばステージ作成の種となるもので，クリアできないステージが作成されるのを防ぐための処理と言える。この処理を行っているのが，switch文のあとのdo～while文じゃ。

そのあとは，指定された数だけ矢印ブロックを配置していく処理になる。最初のfor文で配置するブロック数を指定する。Ksu-1となっているのは，前述した→←ペアのブロックを除いた数になるからじゃな。また，矢印ブロックは，同じ色のペアがないと，ゲームがクリアできないので，最初に配置する色を指定したあと，その色で二つのブロックを配置する処理がされている。そのため二つ目のfor文(変数iが使われている方)が存在しているわけじゃ。

ブロックの配置方法は，乱数で配置する位置を決定し，その位置が空白であるならば，配置するブロックの種類を同じく乱数で決定し，配置していく。ただし，配置する際には，ブロックの進行方向(矢印の方向)にブロックが置かれているかを調べておる。この処理を入れないと，いきなりゲームオーバーになってしまうステージが作成されてしまうのじゃ。この配置処理を行っているのが，if (Bm[x][y] == 0){}内の処理じゃ。

さて，ここまでの解説をリストを照らし合わせながら読んでいる読者は，一つif文を抜かして解説していることに気が付いたのではないかな？そのもう一つのif文(if (Bm[x][y] > 0))は，ステージのブロック配置を修正するもので，実はこの処理がなくてもステージは作成される。しかし，この処理を入れることで，ステージ作成がより効率良く，また内容もより複雑なものになるのじゃ。この部分についての解説は，**リスト2**のコメントおよび**写真2**を参照してほしい。

ステージ作成の最後の処理は，作成

したデータをKm[][]へ代入する処理。ゲーム本編の処理では，ステージデータをKm[][]から参照しておるので，この処理を忘れてはいけないゾ。

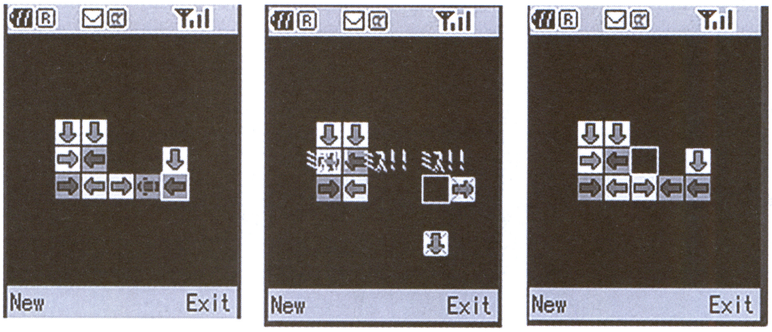
さて，ここまで長くなったが，ステージ作成部分を詳しく解説してみた。パズルゲームのステージを作成する上で，一番注意しなければならないのは，クリアできないステージを作成しないこと。このプログラムでは，特別難しい処理はしていないが，最初に→←ペアを配置するなど，アイデア面での工夫で，この問題をクリアしているのがポイントじゃな。

BEST PROGRAM

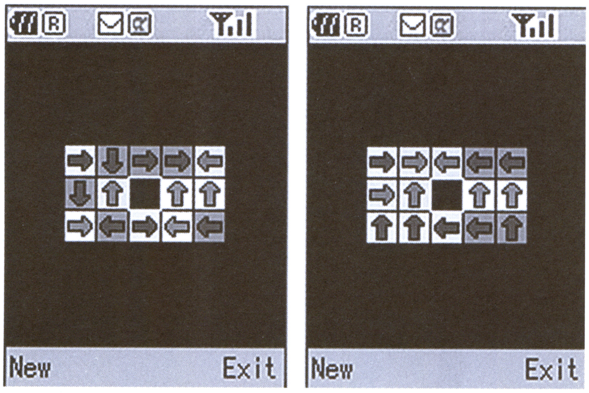
## 改造してみよう

今回行う改造は，すでに行ってしまったブロックの移動処理を一つ前の状態に戻せる機能＝アンドゥ機能を搭載させるというものじゃ。ブロックの移動先を読むのが難しいこのゲームでは，アンドゥ機能をもたせることで，パズルゲームが得意でない人も，ある程度プレイが進むようになると思う。それでは改造方法を紹介していこう。

アンドゥ機能をもたせるには，常に移動前の状態を保存しておくため，その保存用の配列を定義する必要がある。そして，その配列へ移動前のステージデータを入れる処理と，実際にアンドゥ機能を動作させる処理を記述すれば，基本的には，アンドゥ機能は動作するのじゃ。じゃが，気をつけなければならないのは，アンドゥ機能を動かしてはいけないとき(ゲーム開始時など)の判断を行う点じゃな。この点さえ気をつければ問題なくアンドゥ機能を搭載できるはずじゃ。改造リスト(**リスト3**)を参考に，ぜひとも改造を行ってほしい。




◀《写真3》アンドゥ機能の動作例※左がミスする前の状態、真ん中がミス画面、右側がアンドゥ機能で一手前に戻した画面




▲《写真2》配置修正処理の有無の違い
※左側が通常処理，右側が配置修正処理を行っていないもの

《リスト2》ステージ上ブロックの配置修正を行うリスト

```
//ステージ上ブロックの配置修正
//この処理がなくてもステージは作成されるが，この処理を入れることで，
//ステージがより効率良く，また複雑に作成される

//処理内容としては，指定した場所にブロックがあった場合，
//そのブロックの矢印と反対方向へ移動できるかを調べ，
//移動できる場合は，ブロックの移動処理を行う。
//移動後は，この後に記述されているブロックの新規配置処理で，
//元あった場所に，新しいブロックが配置される

if (Bm[x][y] > 0){ //すでにブロックが置かれている場所
        switch (Bm[x][y]){
        //矢印の進行とは逆の移動方向を指定
        case 1: case 5: case 9: case 13:  // ←ブロック
                x2 = 1; y2 = 0; break;
        case 3: case 7: case 11: case 15: // →ブロック
                x2 = -1; y2 = 0; break;
        case 2: case 6: case 10: case 14: // ↑ブロック
                x2 = 0; y2 = 1; break;
        case 4: case 8: case 12: case 16: // ↓ブロック
                x2 = 0; y2 = -1; break;
        }
        //移動先が空白なら，その場所へブロックを移動
        if (Bm[x + x2][y + y2] == 0){
                h = 1;
                Bm[x + x2][y + y2] = Bm[x][y];
                Bm[x][y] = 0;
        }
}
```

《リスト3》アンドゥ機能を付ける改造リスト

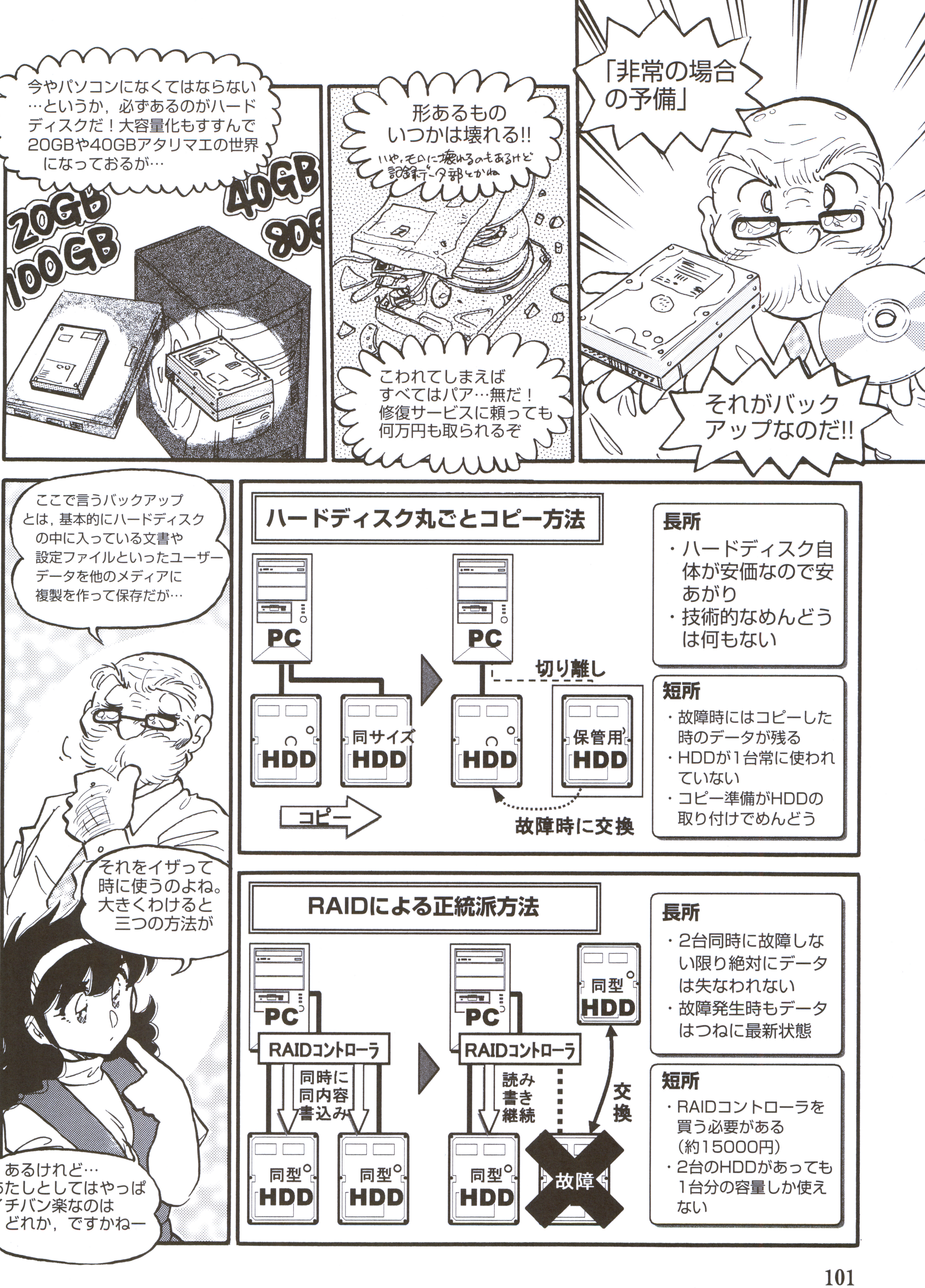
```
//追加するグローバル変数は，以下の2つ
   private int undo_f = 0; // アンドゥを行うかどうかの判断用
   private int undo_stage[][] = new int[9][9]; // 1手前のステージデータ保存用

//追加するメソッド(関数)
   private void undo(int f){
           //f=0:アンドゥを実行　f=1:ステージ内容を記憶

           int i,j;

           if(f==0){
                   for(i=0; i<9; i++){
                   for(j=0; j<9; j++){
                   //ステージデータ(Km)に，1手前のデータ(undo_stage)を
                   //代入していく
                           Km[i][j] = undo_stage[i][j];
                   }
                   }
                   //1度アンドゥを行ったので，再びアンドゥを動作させないようにする
                   undo_f = 0;

                   //ブロック数を修正
                   ksu++;
           }

           else if(f==1){
                   for(i=0; i<9; i++){
                   for(j=0; j<9; j++){
                           //ブロック移動前のステージデータを保存しておく
                           undo_stage[i][j] = Km[i][j];
                   }
                   }
                   //アンドゥが行えることを示す
                   undo_f = 1;
           }
   }

//キー入力部分に追加する記述
           case Display.KEY_POUND: // #キーで動作させる
                   if(undo_f >= 1){ //実行OKなら
                           undo(0); //アンドゥ実行
                           dF = 1;
                           repaint();
                   }
           break;

//paintメソッド内の追加記述
           case 4:
           //2つのコマ消す
                   undo(1); //消去前に，ステージデータ保存を行う
                   g.lock();
           Km[bX][bY] = 0; Km[rX][rY] = 0;
                   //以下省略
```

マンガで学ぶ　**コンピュータの世界**

# パソコンレクチャー ザ・ビギニング

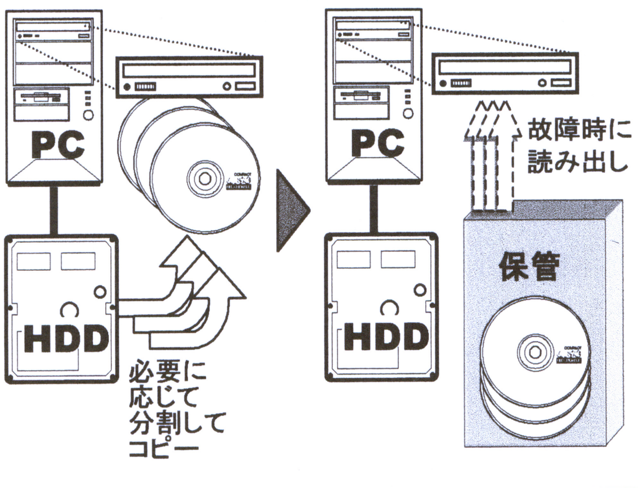
# 「バックアップも忘れないで！」

by くりひろし



今やパソコンになくてはならない…というか，必ずあるのがハードディスクだ！大容量化もすすんで20GBや40GBアタリマエの世界になっておるが…
20GB
100GB
40GB
形あるもの いつかは壊れる!!
いや、モロに壊れるのもあるけど 記録データ部とかね
こわれてしまえば すべてはパア…無だ！ 修復サービスに頼っても 何万円も取られるぞ
「非常の場合の予備」
それがバックアップなのだ!!
ここで言うバックアップとは，基本的にハードディスクの中に入っている文書や設定ファイルといったユーザーデータを他のメディアに複製を作って保存だが…
それをイザって時に使うのよね。大きくわけると三つの方法が
あるけれど… あたしとしてはやっぱイチバン楽なのは どれか，ですかねー
ハードディスク丸ごとコピー方法
PC
HDD
同サイズ HDD
コピー
PC
切り離し
HDD
保管用 HDD
故障時に交換
長所
・ハードディスク自体が安価なので安あがり
・技術的なめんどうは何もない
短所
・故障時にはコピーした時のデータが残る
・HDDが1台常に使われていない
・コピー準備がHDDの取り付けでめんどう
RAIDによる正統派方法
PC
RAIDコントローラ
同時に 同内容 書込み
同型 HDD
同型 HDD
PC
RAIDコントローラ
読み 書き 継続
同型 HDD
故障
交換
同型 HDD
長所
・2台同時に故障しない限り絶対にデータは失なわれない
・故障発生時もデータはつねに最新状態
短所
・RAIDコントローラを買う必要がある（約15000円）
・2台のHDDがあっても1台分の容量しか使えない

## 大容量メディアによる方法




**長所**

- メディアの種類によるが長期保存ができる
- 容量の大きいメディアほど相対的に安価
- メディアはバックアップ以外にも使える

**短所**

- 最近のHDDは容量が大きすぎて，複数枚のメディアが必要になる
- データ書き込みに手間もかかるし時間もかかる

| | メディア | 容量 | 長所 | 短所 | 市場価格 |
|---|---|---|---|---|---|
| CD系 | ・CD-R<br>・CD-RW | CD-R 700MB<br>CD-RW 650MB | ・メディアが安い<br>・標準装備のPCも増えている<br>・24倍速などの高速書き込みも可 | ・容量が今ひとつ少ない<br>・経年劣化や日光などによる劣化がある | CD-R 約60円<br>CD-RW 約350円 |
| MO系 | ・MO<br>・GIGAMO | MO 640MB<br>GIGAMO 2.3GB | ・保存性が非常に高い<br>・CD/DVD系に比べて読み書きがラクラク | ・書き込み速度がが今ひとつ<br>・メディアがやや高め | MO 約600円<br>GIGAMO 約2300円 |
| DVD系 | ・DVD-RAM<br>・DVD-R<br>・DVD-RW | DVD-RAM 9.4GB<br>それ以外 4.7GB | ・とにかく容量がケタちがいで大きい<br>・容量に比べてメディアが安価<br>・DVD-RW系はCD-R/RWとの兼用もある | ・DVD-RAMはホコリ，ヨゴレに弱い<br>・ドライブによってサポートしている方式がちがう（互換性が低い） | DVD-RAM 約3000円<br>DVD-R 約800円<br>DVD-RW 約1200円 |
| ディスク系 | ・ZIP<br>・スーパーディスク | ZIP 250MB<br>スーパー 240MB | ・保存性はいい<br>・スーパーディスクはFDと共用可 | ・メディアがやや高め<br>・メディアが入手しづらい | ZIP 約1200円<br>スーパー 約1000円 |
| テープ系 | ・DAT | 長さに依存<br>〜100 B | ・テープ1本でOKな大容量 | ・保存性が今ひとつ<br>・ドライブは非常に高価<br>・メディアが入手しづらい | 40GBで 約4000円 |
| メモリ系 | ・メモリースティック<br>・SDカード<br>・スマートメディア<br>・CFカード | メディア毎<br>8〜320MB | ・保存性は良い<br>・場所をとらない | ・メディアは高い<br>・静電気等に弱い | 128MBで 約1万円 |
| HD系 | ・マイクロドライブ<br>・モバイルディスク | 1〜5GB | ・保存性は良い<br>・場所をとらない | ・メディアは高い<br>・衝撃に弱い | マイクロドライブ 約5万円 |
| FD系 | ・3.5インチフロッピー | 1.44MB | ・ほとんどのマシンで標準装備 | ・枚数がかさみすぎ非実用的 | 約40円 |

（1枚あたり）

メディアによるバックアップの場合，メディア側の容量がハードディスクの容量アップにおいついていないのが難点だ
容量
HDD容量
メディア容量
年
もう少し正しく言うと，出はじめの新メディアはその時のハードディスクの容量をバックアップするのは可能だが，まだ高価だ——普及して安価になったころにはもう容量負けしておる
つーことは，すべてをバックアップしきるってのはそもそもムリがあるってことね
…というかホントーにバックアップを取る必要のあるものをバックアップしとるのか，と。
OSはそもそもバックアップはいらんな
OSに付属のバックアップ機能はまぁ使うとして…
たとえば，アプリケーションソフトの場合，個人設定ファイルやユーザーファイル以外はバックアップしてもムダになるだけじゃな
そうおもわん人もいるがのう
うーんあとは使い方ですね…メディアの特性を考えてバックアップするのがいいですよね
ま，ふだんからバックアップ用にもなるメディア自体に最初からデータを入れておいて定期的にメディアのバックアップを取るという手もある
文書
メール
MO
画像
CD
動画
DVD
あー，ところで念のために言っておくがいわゆる違法コピーはバックアップなどじゃないからの
違法コピー
うーんやっぱバックアップって大事なのか
この大事なデータ，今度こそバックアップを取らないと
よーし，これで安心
…と，
まてよ，コレが こわれたら困るな
…ようやくこれで安心して…いやいやこれだって壊れるかもしれんな
ううーむ やっぱりバックアップのバックアップをバックアップして…
バックアップうぅぅ
うわああぁぁん 影さんが壊れちゃってるよおぉ
こりゃなおしようが…！
TO BE CONTINUED…

これから始めるブロードバンド活用術

# ADSL HowTo!! 体験記

## 「eAccess」接続サービスでの設定方法

**ADSLやCATV接続サービス・エリアの拡大で全国の主要都市周辺で，高速なインターネット回線が低価格で利用できるようになっています。それに伴い，ブロードバンド回線を使った新たな通信サービスが登場しています。インターネット・フォンもその一つです。今回は，インターネット利用の新技術と「eAccess」接続サービスのセットアップを中心に紹介しましょう。**

### ブロードバンドの新サービス インターネット電話とは？

昨年末に発表があり，今年の3月から始まった，ブロードバンド回線を利用したインターネットの新サービスが，Yahoo! BBが行っているインターネットの電話接続サービスの「BBフォン」です。これまでにも，インターネットを電話回線として利用する試みが行われてきました。

一般に，インターネット・フォン（電話）と呼ばれているものがそれで，「BBフォン」は，このインターネット・フォンの流れをくむ新サービスといってよいでしょう。

インターネットが一般に利用できるようになった当初から，インターネットを経由し音声や映像を双方でリアルタイムに送受信する方式は，いくつか登場しています。しかし，その頃はアナログ・モデムでの接続が主流で，音質や画質はひどいもので，実用レベルにはほど遠いものでした。

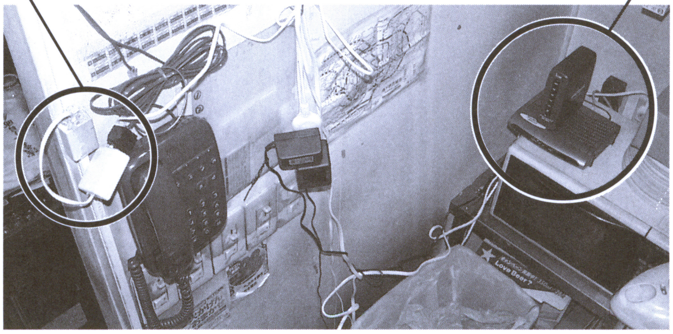
企業などを対象にしたインターネットの電話接続サービスなどもありましたが，導入には特殊な専用装置が必要で，一般ユーザーには手が出せないほど高価なものです。

いまでは，数Mbpsもの高速通信が可

## 実録 アナログモデム通信から8MbpsADSLへの乗り換えにチャレンジ!! eAccess + Nifty ブロードバンド導入てん末レポート Part ②

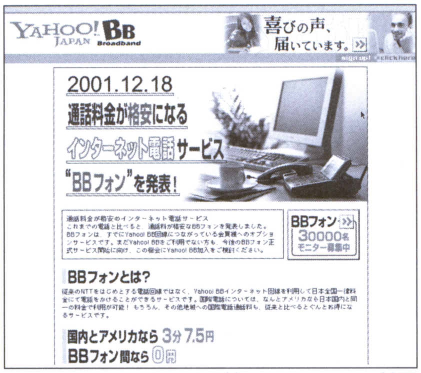
イー・アクセスからADSL開通予定日の2月1日の前日に届いたADSLモデムとスプリッタを早速，電話回線のモジュラー端子に接続し，同梱の接続マニュアルを見ながらモデムの設定を一通り済ませました。さて，開局当日になり，インターネット・エクスプローラを起動し，インターネットへアクセスというところで，接続不可の画面に…。設定をいくら確認しても，間違いはないようです。半日かけケーブルの接続など，思いつくすべての方法を試してみましたが，まったく状況は変わらずじまい。困り果てて，アット・ニフティのサポート窓口へ電話を掛け状況を説明し，オペレータの指示どおり再設定をしてみましたが，接続はできません。結局ニフティからeAccessへ連絡を取ってもらうことに…。(**取材協力・がらん**)

ADSL加入電話回線のモジュラー端子とスプリッタ



eAccessから有償提供のルータ機能付きモデムとテスト用に用意した無線ルータ（今回は使用していない）

**1月31日 スプリッタ・モデムの接続設定**



▲3月にサービスが開始されたADSL利用のインターネット電話「BBフォン」

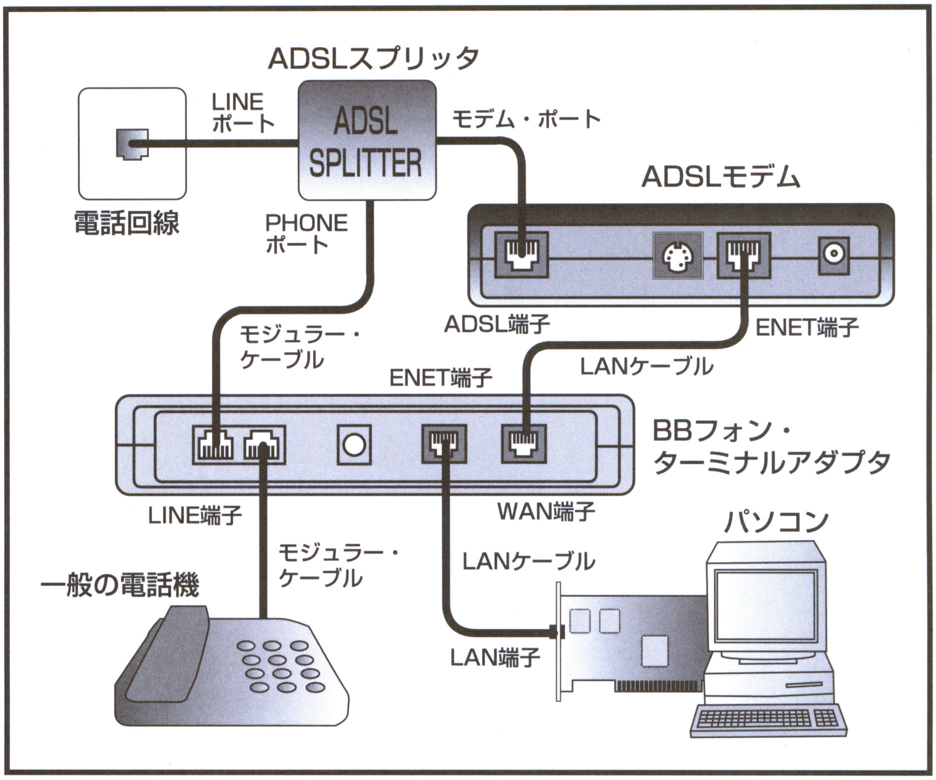
● 「BBフォン」料金

| | |
|---|---|
| 初期登録手数料 | 3,980円 |
| 基本料金 | 390円／月 |
| BBフォン専用<br>接続機器レンタル料 | 140円／月 |

※Yahoo!BBへの入会が必要

● 「BBフォン」通話料金

| | |
|---|---|
| BBフォン同士 | 国内外共無料 |
| BBフォン→<br>国内一般電話 | 7.5円／3分 |
| BBフォン→<br>国外(米国内)一般電話 | 7.5円／3分 |



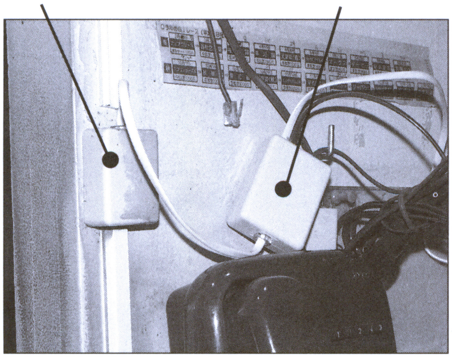
▲Yahoo! BB「BBフォン」の接続方法

能な常時接続のブロードバンドが，極めて安価に利用できるようになり，以前とは比較にならない快適なインターネット環境が整っています。その環境をフルに利用したのが，「BBフォン」なのです。

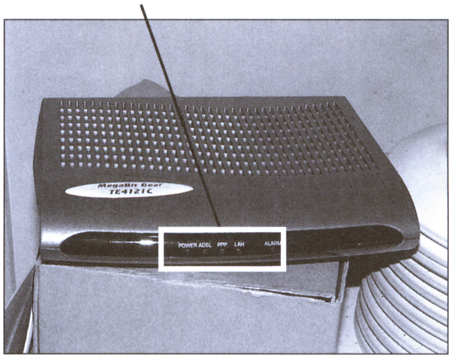
ご存じのように，動画データの送受信に比べ音声のやりとりはデータ容量が少なく，ADSLなどの接続環境でも十分なデータ転送容量が確保できます。そのため一般の電話回線よりも音質の良い明瞭な音声で通話が可能という特徴があります。さらに，転送速度に余裕があるためデータ通信と音声通話を同時に行うことも可能です。

導入にかかわる専用装置も，以前とは比較にならないほど簡略化され，接続は極めて簡単で，安い導入費用で済むのが特徴となっています。

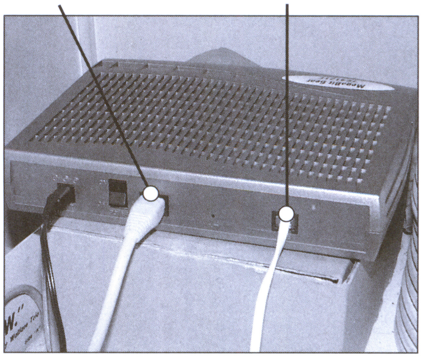
「BBフォン」の利用は，「Yahoo! BB」のADSL接続サービスへの加入が原則となっていますが，「BBフォン」同士の接続であれば，通信料金は無料です。一般加入電話との通話も可能で，こちら



▲電話回線のモジュラー端子から4芯モジュラー・ケーブルでADSLスプリッタに接続。ここからモデムと電話機へ分岐し接続



▲eAccessのモデム。ルータ機能付きで，フロント・パネルには動作チェック用のLEDランプが配置されている



▲モデム背面のケーブル接続の様子。電源端子（ACアダプタ用），電源スイッチ，LAN端子，ライン端子が見える

## サービスエリアを大幅に拡大する ブロードバンド接続の新技術「リーチDSL」に注目!!

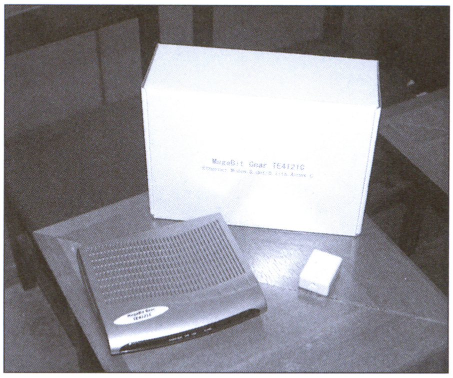
ADSLの欠点といえるのが，規定値に近い転送スピードを確保できるエリアが，アクセス・ポイントの電話局から数kmまでと極めて狭いこと。そのため，ADSL回線を導入したが期待ほどのスピードが得られないトラブルも多いようだ。その解決策として本来の速度を半分程度に制限し，サービスエリアの距離を伸ばす方式が幅広く導入されようとしている。「Yahoo! BB」の「リーチDSL」がそれで，電話回線の周波数帯域を従来より狭く設定し，ISDN回線の干渉を抑え，耐ノイズ性能を高めることでサービスエリアをさらに広くすることができる新技術なのだ。通信速度は上り／下り最大1 Mbps程度で，サービスエリアは5 kmまでのばすことが可能になり，これまでADSLの恩恵にあずかれなかったエリアにもブロードバンド環境がこのリーチDSLで利用可能になる優れた方式といえるだろう。

▲リーチDSLの対応エリアも続々増えている。http://bbpromo.yahoo.co.jp/ で事前に確認可能

は，3分間7.5円の有料となりますが，市内通話，市外通話はもとより外国（米国内）との通話もこの料金で行える破格の通話料金が設定されています。

いずれにしても，長電話派の人には，毎月の電話代請求の金額にビクビクしなくて済む極めてお得なサービスに違いないはずです。また，NTTの電話回線を追加しなくても，プライベートで使える電話が簡単に手に入るメリットも見逃せませんね。

一般の電話回線でも通話料の安さをウリにしている電話接続サービスがあります。しかし，国内外を問わず一律3分7.5円の通話料は，画期的といってもよいでしょう。



▲eAccessから有償（レンタル）提供のルータ機能付きモデム「MegaBit Gear TE4121C」

なお，Yahoo! BB以外のいくつかのインターネット接続サービス会社でも，同様なインターネット電話接続サービスが開始されるようです。こちらも，気になります。

今回のサービス開始に合わせ，「BBフォン」に申し込んでおきましたので，次号で使用感や実際の導入方法を紹介できればと考えていますので，ご期待ください。

### eAccessとYahoo! BBでの設定方法の違い

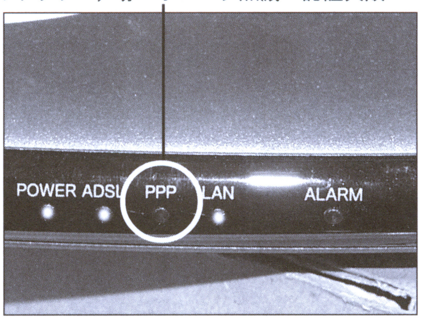
先月号までは，Yahoo! BB接続サービスでのADSL導入方法を中心にしていましたが，今回は「eAccess」（8 Mbpsフルレイト・タイプ）を紹介しましょう。

まず，「eAccess」と「Yahoo! BB」と

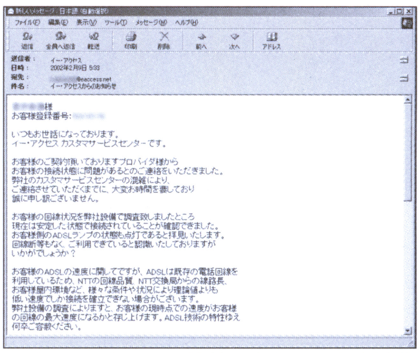
## 実録 「eAccess＋Nifty」ブロードバンド導入てん末レポート②

イー・アクセスからの返事を待ちながら数日間のADSLモデムの動作状態をチェックしていましたが，ADSL，LANのLEDランプは緑色に点灯し，PPPだけが，緑とオレンジ色に点滅を繰り返し，指定どおりの認証IDとパスワードをモデムに設定しているにもかかわらず，認証ができない（？）状態のままで，もちろんインターネットへのアクセスはまったく行えません。イー・アクセスからの連絡がないまま，1週間がたち突然何の問題もなく繋がるようになりました。はっきりした原因はいまだに不明です。

緑点灯：認証ID，パスワードを認証／オレンジ：点滅は認証中，点灯はIPアドレス取得不可／赤点滅：認証前のトラブル／赤-オレンジ点滅：認証失敗



▲ADSL開通日になったが，PPPランプがオレンジと緑色の点滅を繰り返すばかりで，本来の緑色の点灯状態にならない…困った!!

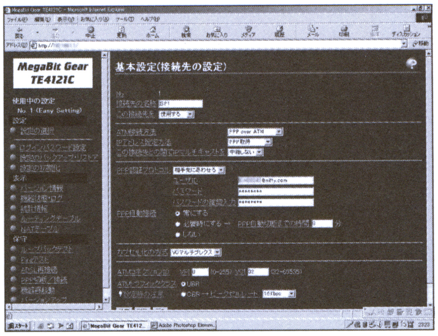


▲イー・アクセスへ接続できないことを連絡し，調査後に送られてきた電子メール。調査の結果問題はないとの連絡内容だが…

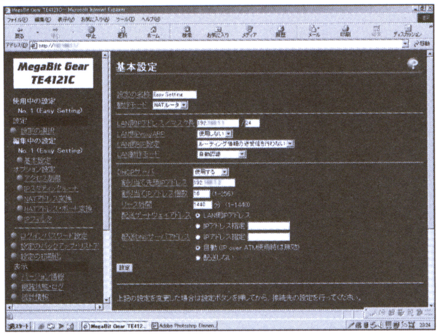
2月1日 ADSL開通予定日

2月9日 やっと開通

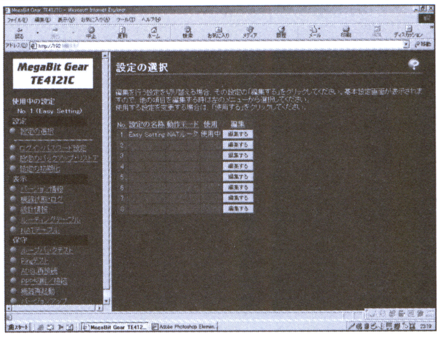
## ●eAccessモデム・ルータの設定項目 ※ウエブ・ブラウザ上から各設定画面にアクセスする



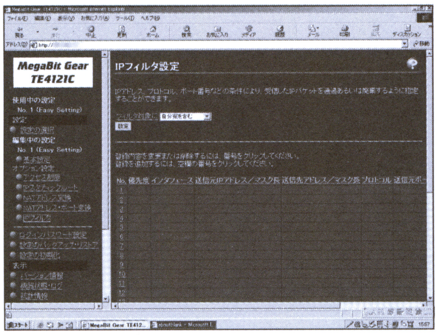

▲「基本設定（接続先の設定）」。PPPoverATM，ユーザID，パスワードを設定



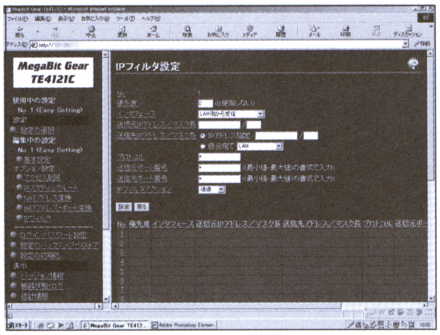

▲「基本設定」。LAN側IPアドレス，DHCPサーバ，DNSサーバなどのルータ機能の設定



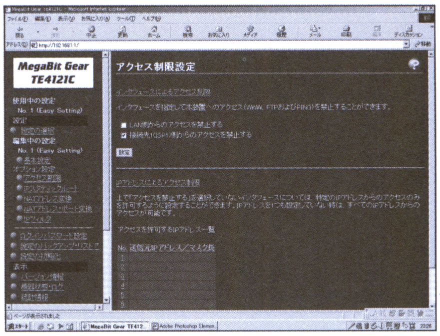

▲「設定の選択」。右のメニューで複数用意した設定の選択を行う




▲「IPフィルタ設定」。外部からADSLモデム設定の改ざんを防ぐセキュリティ機能




▲「IPフィルタ設定」。送信元IPアドレスやプロトコル，通過・非通過などの項目を選択




▲「アクセス制限設定」。ADSLモデムへの外部からのアクセスに条件を付ける

では，ADSLの回線接続方式が異なります。「eAccess」は日本独自のANNEX C方式，「Yahoo! BB」は米国方式のANNEX A方式が元になっています。

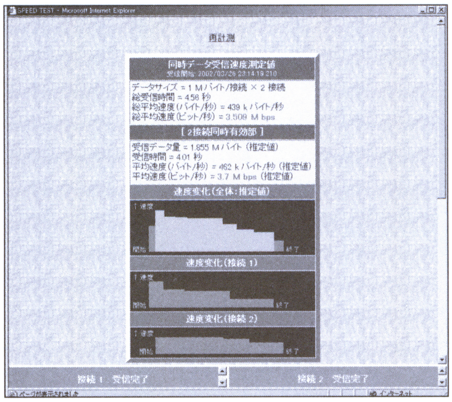
接続方式の違いは，それぞれのADSLモデムが処理するため，通常は利用者が意識しなくても問題はありません。それよりもモデムの設定方法の違いに注意すべきです。というのは，現時点で「Yahoo! BB」は，加入時に有償で提供されるのがモデム（専用）タイプで，「eAccess」では，モデムルータ・タイプだからです。

「Yahoo! BB」のモデム・タイプは，ユーザーからの設定操作はまったくなくADSL回線につなぐだけでパソコン側にインターネット接続に不可欠なIPアドレスが供給されます。モデムルータ・タイプの「eAccess」は，ルータ機能の設定を予め行う必要があります。

また，汎用のモデムルータを利用しているため，接続プロトコルに「eAccess」

PPPのLEDランプが緑色に点灯し，インターネットへのアクセスが可能になった

▲モデムの設定は，最初からまったく変更していないにもかかわらず，突然PPPランプの点滅が緑色の点灯に変わり，インターネットへ繋がるようになった。ADSL工事の不備が原因のように思えるのだが…。インターネット情報によると，良くあることらしい




▲早速，接続速度を調べてみた。平均速度は3.5Mbpsと8 MbpsのフルレイトADSLとしては少々不満もあるが，今後のOSチューンで速度アップを図りたい!!

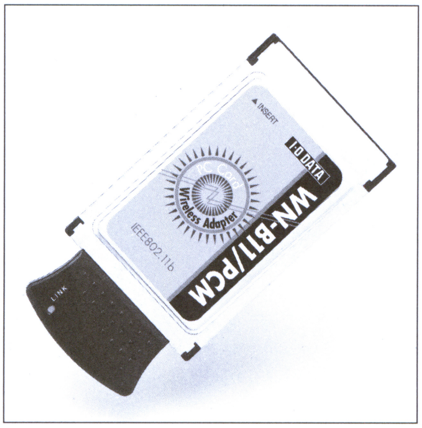
▲ADSL回線が開通したら，次は無線LANでワイヤレスのブロードバンド環境にチャレンジ!! お馴染みの価格.COMで導入機器を選択（以降続く）

## デスクトップ機からノートPCまでサポートするコンパクトな無線LAN機器・シリーズ

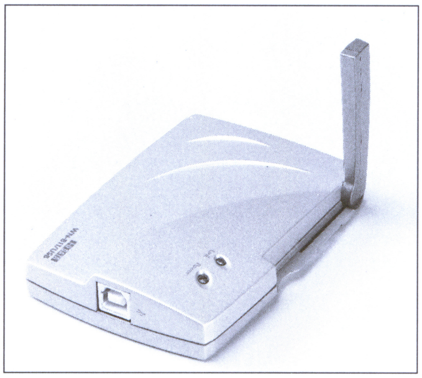
ノート・パソコンで最大11Mbpsの超高速な無線LANシステムが簡単に導入できる注目の製品だ。

お父さんのノート・パソコン，自分のデスクトップ機といったように，自宅で複数のパソコンが使える環境も珍しくなくないようだ。だが，インターネット回線をそれぞれのパソコンから利用するには，ネットワーク・ケーブルを各部屋へ配線しなければならない。ケーブル配線が不要でインターネット接続を実現する便利なツールが，アイ・オー・データ機器の無線LANアクセス・ポイント「WN-B11/AXP」。

ブロードバンド・モデムに接続することで，各パソコンからインターネット接続が可能になるすぐれもの。パソコンのUSB端子に接続する無線LANアダプタ「WN-B11/USB」やPCカード・タイプの「WN-B11/PCM」で，デスクトップ機やノートパソコンからも使える




▲ノートパソコンのカードスロットへの装着で無線LANシステムへアクセス可能になる「WN-B11/PCM」(8,500円)



◀ケーブルが抜き差し自由のUSB端子で無線LANが利用可能になる「WN-B11/USB」(12,000円)。デスクトップのほかノートパソコンからも使える

▲ルータ機能付きモデムのLAN端子への接続で複数のパソコンからワイヤレスで同時にインターネットアクセスが可能になる無線LANアクセス・ポイント「WN-B11/AXP」(19,800円)

で用いられている「PPP over ATM」を選択するなどのユーザー設定を行うことになります。また，「Yahoo! BB」では不要な，ユーザーIDとパスワード設定なども行わなくてはなりません。

### セキュリティ対策は万全 eAccessのルータモデム

こういうと，「Yahoo! BB」のほうが設定が簡単で初心者向きと思われるかもしれませんが，WAN(ワイド・エリア・ネットワーク)側のグローバルIPアドレスがそのままパソコンに割り振られる「Yahoo! BB」に比べ，パソコンへローカルIPアドレスに変換するルータ機能が使える「eAccess」では，ファイアウォールで外部からの不正アクセスをシャットアウトできる強力なセキュリティ機能が利用できるメリットがあります。

▲ルータ機能付きのeAccess用ブロードバンド・モデムと無線LANは最強のインターネット環境の組み合わせ

本格的な利用には少々ネットワークについての専門的な知識が必要になるものの，外部からのアクセス・データを個々のIPアドレスに応じて強制切断したり，許可を行う多機能IPフィルタ機能も用意されています。

ADSL回線でウエブサーバやファイル・サーバをインターネットで公開する場合は，ハッキングなどにさらされる機会も増えます。そのような場合は，このIPフィルタが強力な味方になってくれます。また，イーサネット・ハブを経由することで，複数のパソコンから同時にインターネットが利用できる点もルータ機能付きモデムの特徴といえます。

以上のような機能を「Yahoo! BB」で実現するには，市販のファイアウォール・ソフトやルータ機器などをモデムとは別に用意しなければならず，予定外の出費を強いられることも少なくないようです。

要するに，設定が簡単という理由だけで，接続サービス会社を選ぶと後で苦労されられることもあります。現在または将来のパソコン環境に合わせたインターネット環境が実現できるブロードバンド接続サービスを選びたいものです。

さて，次回は「eAccess」のルータモデムと「Yahoo! BB」のモデムで，複数のパソコンからインターネット利用を可能にするネットワーク接続の実践テクニックと，最近注目を集めている無線LANを使ったブロードバンド接続システムについて紹介します。

(文・福場龍夫)

# もっと身近にプログラミング

チャレンジ!!

## オリジナル・プログラム言語の制作!

### その4 ジャンプには，ついてまわるよコンバータ

**文** まかべひろし

FFXIのβ版で遊んでます。感想は，よくまとまったネットゲームというところでしょうか。PCでネットゲームを遊んでいる人からは，さほど新鮮味はないかなぁ。それより良くできてるのが，Playonlineビュア。近未来的デザインとセンス，あとパッドのわりに良い操作性で，ようやくネットもコンシューマに降りてきたなという感じを受けるのです。

**絵** 桜部のとす

花粉です、花粉。とりあえず何が原因なのか検査とかしてきました〜
・・・杉とか松とかブタクサとかいっぱい反応、とほ〜

catlikeホームページ ▶ http://catlike.marchen.to/

**その昔，バッチファイルというものが存在しました。彼は，プログラムと呼ばれることはありませんでした。なぜなら，バッチには，自由に飛び回るための翼がなかったからです。**

#### ほけー。

春ですよー。あたたかいのですよー。草木が青々として，空気が心地よく新鮮に感じられて，草原でごろんと寝転がって昼寝するには最高の季節なのですよー。そんなわけで，プログラマーといえども，ノートパソコン持って外でお仕事することが最近多いのです。春眠暁を覚えず，家や会社でこもってプログラムしてたら，ついつい眠ってしまいますもんね。外でなら，どんなに気持ちよくったって，いくらなんでも寝てしまうことは，寝てしまうことは・・ぐぅ。

ぽかぽか陽気の日差しを感じながら，こうやって，春の幸せなプログラムはできあがっていくものなのです。

#### 今回は，待望のジャンプ機能搭載だ！

てなわけで，幸せを運ぶベーマガマクロプロセッサ（以下BMP）の制作も，いよいよ終盤に突入です！前回までは，変数処理，演算，判定と作ってきました。これによって，上から下に流れるバッチ的なプログラムを組むことはできるようになりました。こうなると，欲しくなるのは，「下から上に」戻ったり，「まったく違う場所に」ジャンプする機能でしょう。BASICでのGOTO命令が，この機能にあたります。

**ベーマガ・マクロプロセッサ略称「BMP」仕様草案**

★整数値を格納できる変数システム
★演算処理は四則演算対応
★条件分岐にIF文。もちろん入れ子対応
・BASICライクにGOTO命令でジャンプ
・サブルーチン処理用にGOSUB/RETURN
・文字列変数や配列も欲しいところ
★デバッグ用に，文字列を画面出力
・マクロ内にコメントを書けるようにする
★は，実装済み

GOTO命令があれば，プログラムの動作順序は自由自在になります。IF命令と組み合わせることで，FORループのような処理も組むことができるでしょう。「プログラム」が「バッチ

（手続き）」と区別される理由には，ひとえにジャンプ命令の存在があります。バッチは上から下へ流れ作業になっているものですが，プログラムは縦横無尽に飛び回るものです。プログラムがプログラムであるためには，GOTO命令という翼が必要不可欠なのです！

というようにリキんだところで，さて，GOTO命令はどのような書式にしましょうか。元になる命令がBASICに存在しますから，それを踏襲することにします。

BASICのGOTO命令は，

**GOTO 飛び先の行番号**

という書式ですよね。同じように，BMPの書式にのっとると，

**:GO 飛び先**

という形になります。ところで，この「飛び先」は，どのように表すべきでしょうか。

## ラベルコンバータの必要性

プログラムとして実装するのを簡単にする場合，一番楽な方法は，飛び先がスクリプトファイルの先頭から数えて何バイト目かを記述することです。これなら，飛び先の数値をそのまま解析カウンタに代入してやれば済む話です。

しかし，スクリプトを書く側からすると，飛び先を書くたびに先頭から何バイトか，一文字一文字数えるのは，とんでもなく大変な作業です。

それならば，BASICのように行数で考えればどうでしょう。エディタの中には行数を横に表示できるものも多く存在するので，これは意外に簡単かもしれません。

ですが，この方法は，修正を要するプログラムには向いていません。たとえば，一番最初の行に１行，命令を追加したとしましょう。初期化の時に画像を1枚多く読みこむ必要が出たなど，ごくごく当たり前に行われる修正と予測されますが，たったこれだけで，GOTOの飛び先を正しく対応させるためには，すべてのGOTO命令の飛び先行を＋１してやらなくてはならないのです。

BASICがこの方法で破綻しないのは，自分で任意に行番号を設定できるからであり，生のテキストの行をそのまま使用してしまうと，頻繁に変更が起きてしまいます。

そこで，BMPでは，飛び先を「ラベル」という形で明示的に作ってやることで対処したいと思います。書式としては，**リスト1**のようにしてみましょう。

**≪リスト1≫ GO命令の書式**

```
:PRINT "スタート"
:GO 'JUMPLABEL'
:PRINT "ここは表示されないよ"
#JUMPLABEL
:GO 'NEXTLABEL'
#NEXTLABEL
:PRINT "終了"
```

実行すると「スタート」「終了」と表示されて終わり，途中の「ここは表示されないよ」という文章はスキップするようにしたいところです。これは，2行目の「GO 'JUMPLABEL'」の命令を判断し，4行目の「#JUMPLABEL」にジャンプすることを期待しているためです。今までどおり，「命令が来たら実行」ということを考えると，GO命令後のラベル名を検索し，見つかった場所にジャンプすればいいように思えます。とはいえ，このサンプルはわずか7行の短いスクリプトですが，実際ゲームを作るとなると，こんな簡易化してあるスクリプトでさえ，ずいぶんと長いものになることは，容易に予測できるでしょう。GO命令が来るたびにラベルの位置を検索していては，はっきり言って処理の足かせになってしまいます。

そこで，**リスト1**が**リスト2**のように変換され，同時にdata1.のようなデータテーブルももつようにしたいと思います。

**≪リスト2≫ 変換されたリスト**

```
:PRINT "スタート"
:GO 0
:PRINT "ここは表示されないよ"
:GO 1
:PRINT "終了"
```

**《データ１》ジャンプ用データテーブル**

```
int nJumpAdr[ 1024 ];
nJumpAdr[0] = #JUMPLABELのあった位置
nJumpAdr[1] = #NEXTLABELのあった位置
```

このようにすることで，GO命令が来たときには，その後に続く「ラベル番号」を得て，nJumpAdr[ラベル番号]の位置にジャンプすることで，処理時間をさほどかけることなくGO命令を実現することが可能になります。この，**リスト1**から**リスト2+データ1**への変換作業を，スクリプト実行の前に行えばいいのです。この変換処理を行うプログラムを，ラベルコンバータと名付けることにします。

## コンバータを実装しよう！

**リスト1**をもう一度見直してください。コンバータを作る前に，それにかけられるスクリプトはどんな書式であるかを，今一度確認します。まず，

**#JUMPLABEL**

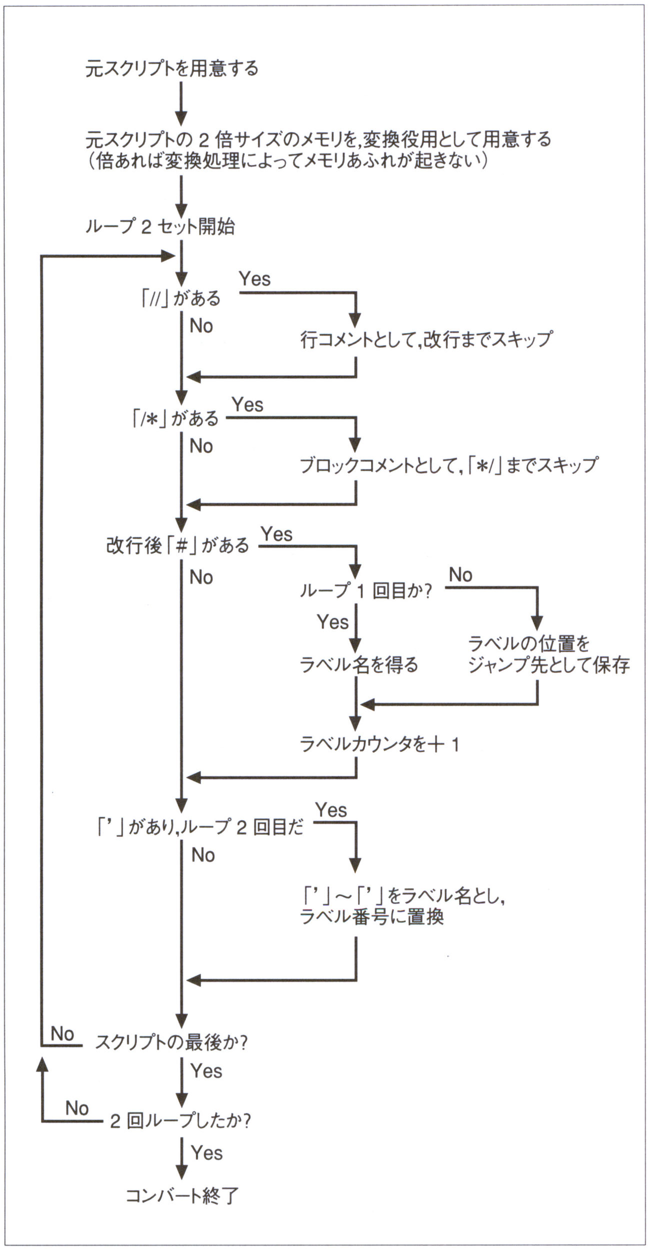
のように，行頭に「#」がついている場所が飛び先のラベルとなり，その後に続いている文字列がラベル名となります。このラベルは，スクリプト中に出てくるたびに，0，1，2・・と昇順に番号を振っていきます。**リスト1**で言うなら，#JUMPLABELは0番，#NEXTLABELは1番となります。

次に，「'」に囲まれた文字列は，番号に置換をしたいラベル名とします。「:GO 'JUMPLABEL'」なら，「:GO 0」とした

いわけです。**リスト1**では飛び先は必ず下方向になっていますが，ループを組む時などは上方向にジャンプする可能性もあります。ラベルの取得とラベル名の置換を同時に行っては，まだ出てきていないラベル名を置換するときに困ってしまいますので，

**（1）スクリプトを上から下に見て，全ラベルを取得する**
**（2）再びスクリプトを上から下に見て，ラベルを置換する**

という，2パス（2回スクリプトを見ていく）構成にしていきましょう。実際のプログラムは，**リスト4**のConverter()が，それにあたります。1文字づつスクリプトを総なめにしていき，改行の後に「#」が来たところをラベルとし，「'」で囲まれたところを2パス目には置換していくというプログラムになっています。




**≪第1図≫ コンバータの流れ図**

これで**リスト1**から**リスト2**へ置換することが可能になりましたが，**リスト2**に付随するデータ1のジャンプ用テーブルはいつ作ればいいのでしょうか。ラベル取得に関しての部分であるので，1パス目に行いたい気もしますが，実際は2パス時にテーブルを作っていくことになります。

だって，考えてもみてください。ジャンプ用テーブルは，1バイトたりともずれてしまってはいけない，精細なデータを要求されているのに，2パス目の時には，スクリプトは「置換作業」によってどんどんずれていってしまう運命にあるのです。置換作業前にラベルの位置を取得してしまっては，2パス時にずれてしまって，実際に使うことはできませんものね。

さて，このコンバータに，もう一つ重要な機能を与えることにしましょう。BMP仕様草案にある「マクロ内にコメントを書けるようにする」も，このコンバータで実装できそうです。つまり，**リスト1**から**リスト2**に変換する時に，コメント部分は削除してしまえばいいのです。

コメントの形はいろいろありますが，個人的には，BASICのREM文以下をコメントとする形より，C言語の「/*」～「*/」間をコメントとする形の方が，より便利に使えると思っています。もちろん，REM文のような1行コメントも欲しいところではあるので，「//」以下改行までをコメントとする形も使います。**リスト3**で実際にコメントを使っているので参照してください。

コンバータの流れとしては，**第1図**のようになります。このコンバータを通すことで，コメントがスクリプトから消え，GO命令のための準備ができるのです。

## GOと一緒にGOSUBもね

ようやくGO命令の実装に入ることができます。が，ここまで出来てしまえばあとは簡単！**リスト4**を見てもらえばわかるとおり，GO命令の実装はわずか3行，

```
int nLabel;
nPnt += GetIntFromString( (char*)&p[nPnt], &nLabel );
nPnt = nJumpAdr[ nLabel ];
```

これだけなのです。すでにおなじみとなった，GetIntFromString()で，「:GO」の後に続く数値を得て，コンバータで作り出したnJumpAdr[]テーブルから，飛び先を得るのです。これによって，スクリプトの参照ポインタであるnPntが，ジャンプ先へと変更されるのです。コンバータの準備で頑張った分，命令自体はかなり単純なものになりました。これなら，GO命令を多用しても実行速度は損なわれないで済みそうです。

この勢いで，仕様草案にある「サブルーチン処理用にGOSUB/RETURN」も片づけてしまいましょう。BASICで言うGOSUB/RETURNは，GOSUBで飛び先にジャンプするのはGOTOと同じですが，RETURNが実行されると，1回前のGOSUBの場所に戻るのが特徴です。これは，いろいろな場所

から呼び出される汎用プログラム……サブルーチンの処理に非常に力を発揮します。

GOTOと違うのは，GOSUB時に「今の位置を保存」し，RETURN時に「前の位置を取り出す」という部分になりそうです。今の位置を指し示しているのはnPntというint型の変数ですから，これを保存するための配列を用意すれば良いのですが，GOSUBは多段で呼び出される可能性があります。つまり，GOSUBで飛んだ先でまた，GOSUBによるジャンプがあるかもしれないのです。この場合，RETURN2回で一番最初の場所に戻るわけですが，そのためには，現在位置を複数保存してやらなくてはいけません。これは，配列変数を利用することで解決します。**リスト4**では，

```
int nGosubStackCnt = 0, nGosubStack[ 64 ];
```

を用意しています。nGosubStack[]には，現在位置の数値が入り，nGosubStackCntには，この配列のどこにアクセスすべきかという配列の添値が保存されます。つまり，1回のGOSUBで，

```
nGosubStack[ nGosubStackCnt ] = nPnt;
nGosubStackCnt++;
```

が実行され，nPntの値がnGosubStack[]に保存されます。これはリスト4では64回まで有効ですが，もっと回数を増やしたい場合には，最初のnGosubStack[]の確保数を増やせばOKです。で，1回のRETURNで，

```
nGosubStackCnt--;
nPnt = nGosubStack[ nGosubStackCnt ];
```

と，nGosubStack[]から値を取り出してきています。この形の場合，GOSUBが複数回実行されると，その都度nGosubStack[]に現在位置が積み重なって保存されていくので，入れ子的なGOSUB命令にも対応することができるわけです。

≪リスト3≫ [macro.txt]

```
}/*
** BMPサンプルリスト
*/

:PRINT "ここから始まります"
// ループを組みます。
:$a = 0
#LOOP
:GS 'SUBPROGRAM'
:$a = $a + 1
:IF( $a < 10 ) { :GO 'LOOP' }
:GO 'END'

// サブルーチンです
#SUBPROGRAM
:PRINT $a
:RT

// ここがプログラム終端となります
#END
```

≪リスト4≫ script.c（差分）

```
#define  LABELMAXNUM    1024    // ラベル数は最大1024ラベル
int nJumpAdr[ LABELMAXNUM ];

// 簡易コンバータ
unsigned char *Converter( unsigned char *pSrc, int *pSize ) {
  int nSize, nCnt, nPass, nLabelCnt, nWriteCnt = 0, nLabelMax = 0;
  char *pDst;
  char nLabelData[ LABELMAXNUM ][ 64 ]; // ラベル名は最大63文字
  nSize = *pSize;
  pDst = malloc( nSize * 2 );
  // 2パス構成なので，2回ループ
  for ( nPass = 0; nPass < 2; nPass++ ) {
    int nRetFlg = 1; // 改行フラグ。改行が直前にあったら1を入れます
    nCnt = 0;
    nLabelCnt = 0;
    // このループでスクリプト全体をサーチします
    while( nCnt < nSize ) {
      if ( IsKanji( pSrc[ nCnt ] ) ) {
        // 全角文字の場合，スキップ
        if ( nPass == 1 ) {
          pDst[ nWriteCnt + 0 ] = pSrc[ nCnt + 0 ];
          pDst[ nWriteCnt + 1 ] = pSrc[ nCnt + 1 ];
          nWriteCnt += 2;
        }
        nCnt += 2;
      } else {
        // 半角文字の場合，解析に入る
        if ( pSrc[ nCnt + 0 ] == '/' && pSrc[ nCnt + 1 ] == '/' ) {
          // 「//」から改行までをコメントとして飛ばす
          nCnt += 2;
          while( !( pSrc[ nCnt + 0 ] == 0x0d && pSrc[ nCnt + 1 ] == 0x0a )
) {
            if ( IsKanji( pSrc[ nCnt ] ) ) { nCnt += 2; } else { nCnt++; }
          }
        } else if ( pSrc[ nCnt + 0 ] == '/' && pSrc[ nCnt + 1 ] == '*' ) {
          // 「/*」から「*/」までをコメントとして飛ばす
          nCnt += 2;
          while( !( pSrc[ nCnt + 0 ] == '*' && pSrc[ nCnt + 1 ] == '/' ) ) {
            if ( IsKanji( pSrc[ nCnt ] ) ) { nCnt += 2; } else { nCnt++; }
          }
          nCnt += 2;
        } else if ( nRetFlg && pSrc[ nCnt ] == '#' ) {
          // 行頭に#があったので，ラベルと見なす
          nCnt++;
          if ( nPass == 0 ) {
            // 最初(pass1)。ラベル名を得てカウンタを進める
            int i = 0;
            while( !( pSrc[ nCnt + 0 ] == 0x0d && pSrc[ nCnt + 1 ] == 0x0a
) ) {
              nLabelData[ nLabelCnt ][ i ] = pSrc[ nCnt ]; i++; nCnt++;
            }
            nLabelData[ nLabelCnt ][ i ] = '¥0';
          } else {
            // 2回目(pass2)。ラベルのある場所をジャンプ先として保存する
            nJumpAdr[ nLabelCnt ] = nWriteCnt;
            while( !( pSrc[ nCnt + 0 ] == 0x0d && pSrc[ nCnt + 1 ] == 0x0a
) ) {
              nCnt++;
            }
          }
          nLabelCnt++;
        } else if ( pSrc[ nCnt ] == '¥'' ) {
          // 「'」があったので，そこから「'」までをラベルと認識して置換。
          int i = 0;
          char szLabel[ 64 ]; // ラベル名は最大63文字
          nCnt++;
          // ラベル名を得る
          while( pSrc[ nCnt ] != '¥'' ) {
            szLabel[ i ] = pSrc[ nCnt ]; i++; nCnt++;
          }
          szLabel[ i ] = '¥0'; nCnt++;
          if ( nPass == 1 ) {
            // 2回目のみ(pass2)。ラベル名をサーチ
            for ( i = 0; i < nLabelMax; i++ ) {
              if ( strcmp( nLabelData[i], szLabel ) == 0 ) { break; }
            }
            if ( i != nLabelMax ) {
```

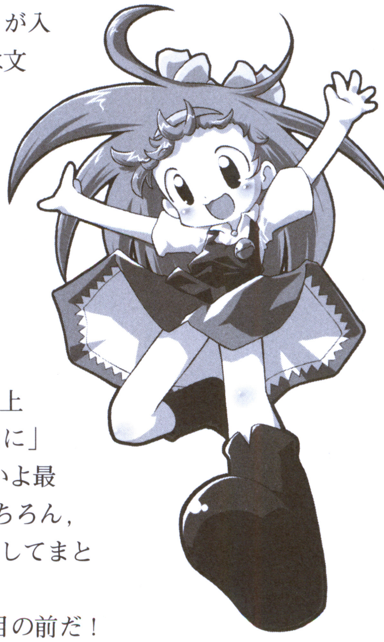
つぎのページに続く

```
          // 見つかったので置き換えて，変換後メモリに書き出し
          char sz[ 32 ];
          sprintf( sz, "%d", i );
          memcpy( &pDst[ nWriteCnt ], sz, strlen( sz ) );
          nWriteCnt += strlen( sz );
        } else {
          // 見つからなかった，エラー
          printf( "error: label not found. %s¥n", szLabel );
        }
      }
    }
    // 改行チェック
    if ( pSrc[ nCnt + 0 ] == 0x0d && pSrc[ nCnt + 1 ] == 0x0a ) {
      nRetFlg = 1; nCnt += 2;
    } else {
      if ( nPass == 1 ) {
        // 2回目のみ(pass2)。変換後メモリに書き出し
        pDst[ nWriteCnt ] = pSrc[ nCnt ]; nWriteCnt++;
      }
      nRetFlg = 0; nCnt ++;
    }
   }
  }
  nLabelMax = nLabelCnt;
 }
 *pSize = nCnt;
 return pDst;
}


// プログラムはここから始まる
void main() {
 FILE *fp; int nSize;
 unsigned char *p, *pWork;
 int nPnt = 0;
 // :GSサブルーチンジャンプは64階層まで可能
 int nGosubStackCnt = 0, nGosubStack[ 64 ];

 // 初期化
 ValueSysInit();

 // マクロファイルをメモリに読み込む
 fp = fopen( "macro.txt", "rb" );
 fseek( fp, 0, SEEK_END ); nSize = ftell( fp ); fseek( fp, 0, SEEK_SET
);
 p = ( unsigned char* )malloc( nSize + 1);
 fread( p, 1, nSize, fp );
 fclose( fp );

 // 読んだデータの終端に¥0をつける
 p[ nSize ] = '¥0';

 // converter //////////////////////////////////////////////////////////////////
 pWork = Converter( p, &nSize );
 free( p ); p = pWork;

 // interpriter /////////////////////////////////////////////////////////////////
 while( nPnt < nSize ) {
  if ( IsKanji( p[nPnt] ) ) {
   // SHIFT-JIS漢字コードなので2byte進める
   nPnt += 2;
  } else {
   if ( p[nPnt] == ':' ) {
    // コマンドマーク有
    nPnt++;
    if ( p[nPnt] == '$' ) {
     // 変数代入
     //
     // 前回と変わらないので省略
     //
    } else {
     // コマンド処理
     char szCmd[32]; int i = 0;
     while( ( 'A' <= p[nPnt] && p[nPnt] <= 'Z' )
      || ( 'a' <= p[nPnt] && p[nPnt] <= 'z' ) ) {
      szCmd[i] = p[nPnt]; nPnt++; i++;
     }
     szCmd[i] = '¥0';
     if ( strcmpi( szCmd, "PRINT" ) == 0 ) {
```

```
      //
      // 前回と変わらないので省略
      //
     } else if ( strcmpi( szCmd, "IF" ) == 0 ) {
      //
      // 前回と変わらないので省略
      //
     } else if ( strcmpi( szCmd, "GO" ) == 0 ) {
      int nLabel;
      nPnt += GetIntFromString( (char*)&p[nPnt], &nLabel );
      nPnt = nJumpAdr[ nLabel ];
     } else if ( strcmpi( szCmd, "GS" ) == 0 ) {
      int nLabel;
      nPnt += GetIntFromString( (char*)&p[nPnt], &nLabel );
      nGosubStack[ nGosubStackCnt ] = nPnt;
      nGosubStackCnt++;
      nPnt = nJumpAdr[ nLabel ];
     } else if ( strcmpi( szCmd, "RT" ) == 0 ) {
      nGosubStackCnt--;
      nPnt = nGosubStack[ nGosubStackCnt ];
     }
    }
   } else if ( p[nPnt] == '}' ) {
    //
    // 前回と変わらないので省略
    //
   } else {
    // コマンドでないので1byte進める
    nPnt++;
   }
  }
 }

 // 解放
 ValueSysRelease();
 free( p );
}
```

## 待て、次号！

実は今回のコンバータ，一つ問題があるのです。それは，:PRINT命令のような「文字列」を渡すものの中に，ラベル置換用の「!」や，コメント用の「/*」「*/」「//」が入っていた場合。これは文字列として認識すべきですが，このままではコメントや置換処理に入ってしまいます。コンバータの方で回避策を考えてみてください。



さて，次回。2年以上続いた「もっと身近に」の連載ですが，いよいよ最終回を迎えます。もちろん，BMPも，次回で形としてまとまるのです！

さぁ，野望達成は目の前だ！

(無　料)
フリーソフト

# Hot Soup Processor HSP 完全マスター

# 楽々3Dプログラミング！

**第4回**

**監修：おにたま**

**講師：悠黒喧史**

いよいよ実践篇である。甘えは許されない。失敗は即，死につながると知れ。NINJAの掟は無情なのだ。3Dゲイム如きが何ほどのものか。ふ，たわいない。

## 「MXモデルで ゲーム制作(1)」

**今回のポイント**

**オブジェクトに属性を設定する**

### ゲームのイメージ

さて，いよいよ実践篇ということで，実際に3Dゲームを制作する過程を解説してゆきます。

3Dゲームでは，制作するゲームのイメージをきちんと立体的にとらえて考えておくことが大切です（**第1図**）。

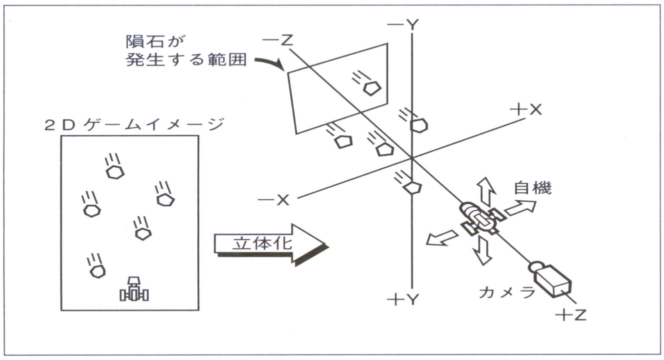
### ゲームのルール

時間とともに得点が加算されてゆき，隕石の数も増えてゆく。

隕石にぶつかるとシールドエネルギーが減り，0のときにぶつかるとゲームオーバーになる。

シンプルなゲームですが，3Dで作る場合，平面的な2Dのゲームよりもプログラミングの手間は多くなります。なめてかからないようにしましょう。

ちなみに筆者の大好きな藤子F作品『モジャ公』の一話「アステロイド・ラリー」からイメージしたゲームです。ポンコツ気味のロケットデザインに，そのあたりのコンセプトが現われているかと(笑)。

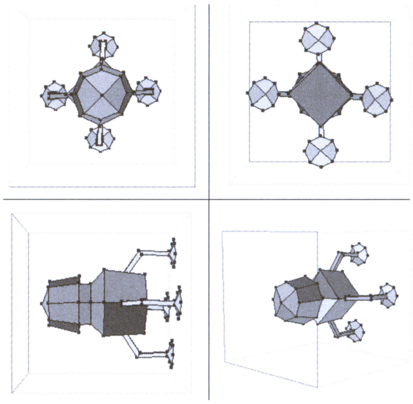



◀《第1図》ロケットで，無数の隕石が飛んでくるアステロイドをくぐりぬけてゆくゲーム

### ゲームに使うMXモデルを作る

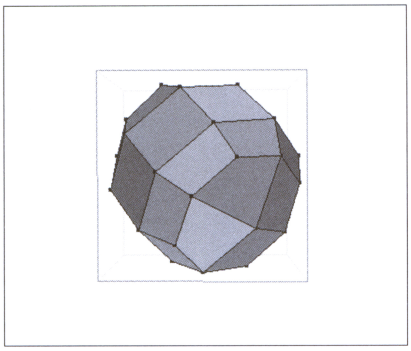
次のようなモデルを作成して用意してください。デザインはともかく，大体のサイズが合っていればOKです。また，MX形式の3Dモデルデータが作れれば，どんなツールを使ってもかまいません。ちなみに筆者は，フリーのモデリングソフト「3DACE」で作成しました。

**ロケット(自機)**
**MXモデルデータ**
myship.mx

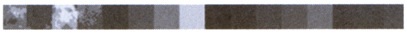


**貼り付けるテクスチャ**
myship.bmp

**隕石**
MXモデルデータ
asteroid.mx



**貼り付けるテクスチャ**
asteroid.bmp



## 自機の移動プログラム

まず，自機のモデルを登録，オブジェクトを作成し，自機を移動させるプログラムから組んでみましょう。

**自機の移動(リスト1)**

### リスト1の解説

**mxload**命令でMXデータ**myship.mx**を読み込み，モデルを登録，自機オブジェクトを作成します。

自機の座標は，ゲームのイメージどおりに中心点より後方(+Z方向)に置き，カメラはさらにその後方に配置しています。

自機の角度は，カメラにおしりを向ける状態にするので，X軸の角度値を-3.14としています。これで180度回転し，ちょうど反対向きになります。

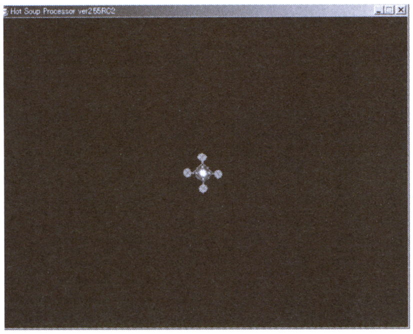
メインルーチンでは，まず，自機のMOC座標を変数poに得ます。

押されたカーソルキーに応じ，変数po.0(X座標値)とpo.1(Y座標値)の内容を加減算し，変数poの内容を自機のMOC座標に戻します。

**hgdraw**命令でウィンドウにオブジェクトを全描画し，hgsync命令で画面を更新します。

ラベル「*main」に戻って，メインルーチンの処理をくりかえします。

## リスト1の実行画面



▲カーソルキーで上下左右に移動します。Escでプログラムを終了します。

**《リスト1》自機の移動プログラムのリスト**

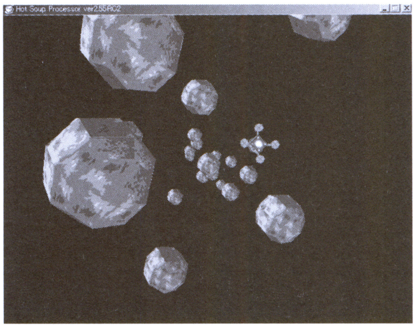
```
#include "hgimg.as"

        hgini                                           ; HGIMGの初期化
; ------------------------------------------------------------------------
        mxload "myship" :mdmy=stat                      ; 自機ﾓﾃﾞﾙﾌｧｲﾙ読込み
; ------------------------------------------------------------------------
        regobj obmy,mdmy                                ; ｵﾌﾞｼﾞｪｸﾄ作成
        selang obmy :objsetf3 -3.1416f,0.0f,0.0f        ; 角度ｾｯﾄ
        selpos obmy :objset3 0,0,50                     ; 座標ｾｯﾄ
; ------------------------------------------------------------------------
        selcpos :objset3 0,0,80                         ; ｶﾒﾗ座標
; ------------------------------------------------------------------------
        gsel 0
*main
        stick ky,255                                    ; ｷｰｽｷｬﾝ

        selpos obmy :objgetfv po                        ; MOC座標を変数po.0～2に得る

        if ky&1 :fsub po.0,0.2f                         ; X軸移動
        if ky&4 :fadd po.0,0.2f                         ;
        if ky&2 :fsub po.1,0.2f                         ; Y軸移動
        if ky&8 :fadd po.1,0.2f                         ;

        objsetfv po                                     ; 変数po.0～2の内容をMOC座標に戻す

        hgdraw :hgsync 15                               ; 描画 / ｳｪｲﾄ

        if ky!128 :goto *main                           ; [Esc]を押すまでﾙｰﾌﾟ

        hgbye                                           ; HGIMGを解放
```

## 隕石をどんどん発生させる

次は，隕石をどんどん発生させて，こちらに向かって飛んでくるようにします。

**隕石の発生(リスト2)**

## リスト2の実行画面

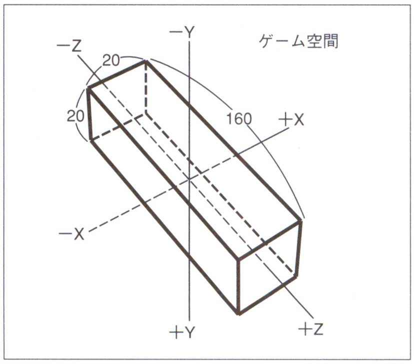


▲カーソルキーで上下左右に移動します。Escでプログラムを終了します。

### リスト2の解説

これは，**リスト1**にいくつかの処理を加えたものです。追加された部分について解説します。

**setborder**命令は，オブジェクトの有効範囲を指定します。ここでは，座標(0,0,0)を中心とする20×20×160の範囲をゲーム空間として設定しています。つまり，座標値では(-10,-10,-80)～(10,10,80)の範囲内ということになります。




《第2図》ゲーム空間(オブジェクトの有効範囲)

### 隕石オブジェクトを作成する

隕石の発生というのは，隕石オブジェクトを作成することです。

隕石の発生タイミングには，乱数を使って，メインループごとに1/9の確率で，隕石オブジェクトを作成するようにしています。

《リスト2》隕石の発生を加えたプログラム・リスト

```
#include "hgimg.as"

        hgini                                               ; HGIMGの初期化
; ------------------------------------------------------------------------
        mxload "myship" :mdmy=stat                          ; 自機ﾓﾃﾞﾙﾌｧｲﾙ読込み
        mxload "asteroid" :mdas=stat                        ; 隕石ﾓﾃﾞﾙﾌｧｲﾙ読込み
; ------------------------------------------------------------------------
        regobj obmy,mdmy,OBJ_FLIP|OBJ_BORDER                ; ｵﾌﾞｼﾞｪｸﾄ作成
        selang obmy :objsetf3 -3.14f,0.0f,0.0f              ; 角度ｾｯﾄ
        selpos obmy :objset3 0,0,50                         ; 座標ｾｯﾄ
; ------------------------------------------------------------------------
        selcpos :objset3 0,0,80                             ; ｶﾒﾗ座標
; ------------------------------------------------------------------------
        setborder 20.0f,20.0f,160.0f                        ; ｵﾌﾞｼﾞｪｸﾄの有効ｴﾘｱを設定
; ------------------------------------------------------------------------
        gsel 0
*main
        stick ky,255                                        ; ｷｰｽｷｬﾝ

        selpos obmy :objgetfv po                            ; po ← 自機MOC座標

        if ky&1 :fsub po.0,0.2f                             ; X軸移動
        if ky&4 :fadd po.0,0.2f                             ;
        if ky&2 :fsub po.1,0.2f                             ; Y軸移動
        if ky&8 :fadd po.1,0.2f                             ;

        objsetfv po                                         ; 自機MOC座標 ← po

        rnd r,9
        if r=0 { regobj obas,mdas,OBJ_MOVE|OBJ_BORDER       ; 隕石ｵﾌﾞｼﾞｪｸﾄ作成
                rnd x,20 :rnd y,20                          ;
                selpos obas :objset3 x-10,y-10,-80          ; MOC座標ｾｯﾄ
                seldir obas :objset3 0,0,1                  ; MOC移動量ｾｯﾄ
        }

        hgdraw :hgsync 15                                   ; 描画 / ｳｪｲﾄ

        if ky!128 :goto *main                               ; [Esc]を押すまでﾙｰﾌﾟ

        hgbye                                               ; HGIMGを解放
```

オブジェクトは，作成時にさまざまな属性を設定することができます。ここでは，隕石オブジェクトの処理に必要な**OBJ_MOVE**と**OBJ_BORDER**という二つの属性を設定しています。

**OBJ_MOVE**属性を設定すると，**MOC**移動量が有効になります。MOC移動量を有効にすると，オブジェクトはプログラム処理中で，毎回，MOC座標に値を加減算しなくても，あらかじめ指定されたMOC移動量に応じて勝手に移動してくれるようになります。

**OBJ_BORDER**属性を設定すると，そのオブジェクトは，**setborder**命令の設定範囲外(ゲーム空間の外)に出たときに自動的に消去処理されるようになります。

**オブジェクトの属性**

OBJ_HIDE
　非表示(画面から消す)

OBJ_TREE
　木属性(Y軸のみ正面を向く)

OBJ_XFRONT
　正面属性(常に画面に正面を向く)

OBJ_UVANIM
　UVアニメーションを行なう

OBJ_UVANIM_1SHOT
　UVアニメーション表示後消滅する

OBJ_MOVE
　XYZ移動量を有効にする

OBJ_FLIP
　ボーダー領域で反転する

OBJ_BORDER
　ボーダー領域を有効にする

OBJ_2D
　2D表示オブジェクトとして扱う

OBJ_TIMER
　消滅タイマーを有効にする

OBJ_WIPEBOM
　ボーダー領域で爆発オブジェクトに変わる

OBJ_NOSORT
　Zソートを無効にする

OBJ_GRAVITY
　重力制御を有効にする

OBJ_SKY
　常に一番奥に描かれる

OBJ_GROUND
　地面オブジェクトとして描かれる

隕石オブジェクトを作成したら，乱数でX,Y座標(X,Y,-80)を初期設定します。隕石を発生させる範囲は(-10,-10,-80)～(10,10,-80)，20×20の範囲内としています。

seldir命令で，OBJ_MOVEで有効にしたMOC移動量を指定します。ここでは，+Z方向に1進むという指定にしています。ちなみに，このMOC値を変更すると隕石の向かってくる速度や向きも変わります。例えば，2とすると隕石の進むスピードが倍になります。

**使用した命令**

setborder
　オブジェクトの有効範囲を設定

seldir
　指定されたオブジェクトのMOC移動量を対象にする

見た目は早くも3Dゲームっぽくなってきたようです。ただ，まだ隕石は，自機を素通りしますし，遊べる状態…ゲームにはなっていません。

次回は，ゲームのルールにあたるコリジョン(当たり判定)処理や，スコア表示などの部分をプログラムに追加して，ゲームとして遊べるカタチに作り上げてゆきます。お楽しみに(^^。

## HGIMG-FAQ

**質問**

3D格闘ゲームを作ることは可能でしょうか？例えば，手足の当たり判定など，どうやったら良いのか，わかりません。

**答え**

結論から言うと，可能です。**HGIMG**は，それだけの機能をそなえています。

ただ，3D格闘ゲームを作るには，**HGIMG**の3D機能そのものよりも，モデリングデータやモーションデータの作成に相当の手間がかかると思います。

あとは，プログラマーの腕次第といったところでしょう。

コリジョン(当たり判定)には非表示のオブジェクトを使って，技の出ている間だけ判定したい部分に重ねればいいんじゃないかな。

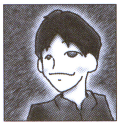
教科「情報」など授業にも使える！

# Visual Basic アルゴリズム講座

文・プログラム
／東 幸太

## 第2回 速度の計算をしてみよう

**冬のオリンピックではスピードスケートやアルペンスキーの滑降，各種そりなどで速度が競われました。速度というのは単位時間当たりの移動量のことで，大きいほど「速く」動いていることになります。今回は，この速度をVisual Basicを使って計算してみます。**



年度の区切りは年の区切りと異なり，パソコン的には何も影響はない。けれど，心機一転を図るには良いチャンスと言える。新しい環境になって心機一転を図っている人が多いと，環境が変わらない筆者も何か新しいことを始めようか，と思えてくる。

### 速度の表現

人間が普通に歩く速度は時速4km，自転車は時速10kmから20km，自動車や電車は時速50km程度・・・という具合に，私たちは普段から速度について表現しています。しかし，通常は「時速」という言葉を抜いてしまって，「20キロ」というような表現をしています。

速度は単位時間当たりの移動距離のことなので，20キロだけでは移動距離しか示されておらず，20キロをどれぐらいの時間で移動したかがわかりません。よって，厳密には「時速」や「秒速」といった言葉をつけなければなりません。

時速ｘキロ，と表現される速度は，その速度を保ったまま移動すると１時間後にｘキロメートルの距離を移動する速度です（第１図）。60kmの距離を４時間かけて移動する速度は，

$$\frac{60\text{km}}{4\text{時間}} = 15\text{km/h}$$

となって時速15kmになります。時間を示す英単語はhourなので，一般的に時速の単位は「km/h」（キロメートル・パー・アワー）で表現します。距離を時間で割った単位になるわけです。

時速は１時間当たりの移動距離に対応していますが，これを１秒にしたのが秒速です。名前のとおり，１秒当たりの移動距離で表現します。こちらは，風の強さを表現する際などに利用され，秒速15mの風，といった具合になります。秒速15mということは，時速に換算すると

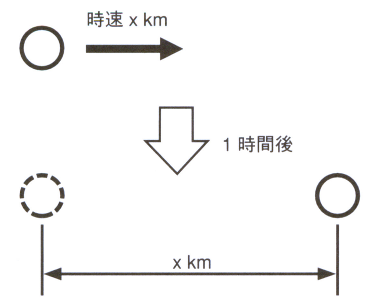
15m/sec × 3600秒 ＝ 54000m/h ＝ 54km/h

となり，時速54kmに相当します。中心付近の最大風速が45mという台風の場合，中心付近では時速150kmを超える速度で風が吹いているというわけです。

なお，km/hに対して，秒速ではm/sec（メートル・パー・セコンド（セック））が単位として利用されます。基準となる時間が１時間なのか，１秒なのか，あるいは距離の単位がmなのかkmなのかによって，同じ速度でも数値が変わってくるわけです。

### 速度の計算

移動距離と移動にかかった時間がわかれば，速度が計算できます。リスト１は



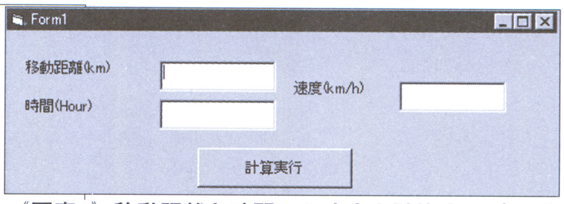

《第１図》時速の定義

《リスト1》速度を計算するプログラム・リスト

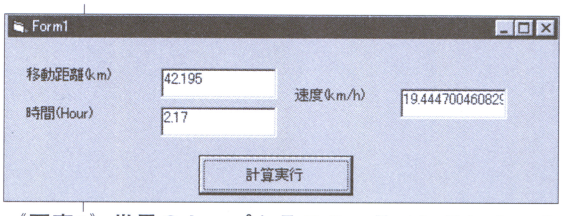
```
VERSION 5.00
Begin VB.Form Form1
   Caption         =   "Form1"
   ClientHeight    =   2028
   ClientLeft      =   48
   ClientTop       =   324
   ClientWidth     =   6888
   LinkTopic       =   "Form1"
   ScaleHeight     =   2028
   ScaleWidth      =   6888
   StartUpPosition =   3  'Windows の既定値
   Begin VB.CommandButton Command1
      Caption         =   "Command1"
      Height          =   492
      Left            =   2400
      TabIndex        =   5
      Top             =   1440
```

つぎのページへつづく

```
      Width         =  1932
   End
   Begin VB.TextBox Text3
      Height        =  372
      Left          =  4920
      TabIndex      =  2
      Text          =  "Text3"
      Top           =  600
      Width         =  1332
   End
   Begin VB.TextBox Text2
      Height        =  372
      Left          =  1920
      TabIndex      =  1
      Text          =  "Text2"
      Top           =  840
      Width         =  1452
   End
   Begin VB.TextBox Text1
      Height        =  372
      Left          =  1920
      TabIndex      =  0
      Text          =  "Text1"
      Top           =  360
      Width         =  1452
   End
   Begin VB.Label Label3
      Caption       =  "Label3"
      Height        =  372
      Left          =  3600
      TabIndex      =  6
      Top           =  600
      Width         =  1092
   End
   Begin VB.Label Label2
      Caption       =  "Label2"
      Height        =  372
      Left          =  240
      TabIndex      =  4
      Top           =  840
      Width         =  1452
   End
   Begin VB.Label Label1
      Caption       =  "Label1"
      Height        =  372
      Left          =  240
      TabIndex      =  3
      Top           =  360
      Width         =  1452
   End
End
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Private Sub Command1_Click()
   Dim dist#, ti#, sp#
   dist = Text1.Text 'テキストボックスに入力された距離を調べる
   ti = Text2.Text 'テキストボックスに入力された時間を調べる
   sp = dist / ti '速度の計算
   Text3.Text = sp '計算結果をテキストボックスに代入
End Sub

Private Sub Form_Load()
   Command1.Caption = "計算実行" 'キャプションの設定
   Label1.Caption = "移動距離(km)"
   Label2.Caption = "時間(Hour)"
   Label3.Caption = "速度(km/h)"
   Text1.Text = "" 'テキストボックスのクリア
   Text2.Text = ""
   Text3.Text = ""
End Sub
```




《写真1》移動距離と時間から速度を計算するプログラムの実行画面

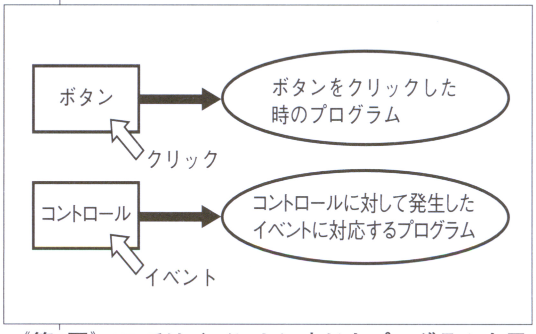



《写真2》世界のトップクラスのマラソンランナーの速度の計算結果

Visual Basicで速度を計算するプログラムです。実行すると，**写真1**のような画面が表示されます。移動距離と時間をセットしてから，計算実行ボタンを押すと，速度がkm/hの単位で表示されます。

世界トップクラスのマラソンランナーは，42.195kmを2時間10分程度で走りきります。**リスト1**を実行し，移動距離を42.195，時間を2.17とすると，19.4という結果が表示されます（**写真2**）。時速20km/hちかい，自転車程度の速度で走っていることになります。

VisualBasicでは，ボタンをクリックしたりすることによるイベントに応じて，プログラムを用意する形でプログラミングを行います（**第2図**）。イベントによってプログラムの流れがコントロールされていくわけです。**リスト1**では，起動時にコントロールが配置されたフォームがロードされた時に発生するイベント(Form_Load)と，計算実行ボタンを押した時に発生するイベント（Command1_Click）について，それぞれプログラムを用意しています。




《第2図》VBではイベントに応じたプログラムを用意していくことで目的の処理を実行させる

なお，この二つ以外にも常時さまざまなイベントが発生しているのですが，プログラムで記述されてないイベントについては，何も実行しないか，Windows側が適切な処理を行うようになっています。よって，必要なイベントについてだけプログラムを用意すればよいことになります。

Form_Loadでは，各コントロールのキャプションや内容を設定することで，**写真1**のような画面が表示されるようになっています。Command1やLabel1といったコントロールのCaptionプロパティを設定することで，コントロールの表示文字列が変わります。この他にも多くのプロパティがあり，大きさや色なども設定することができます。これらのプロパティはフォームエディタで指定しておくことも可能です。

Command1_Clickでは，速度の計算を行っています。このイベントが発生した段階では，ユーザーが移動距離と時間を対応するテキストボックスに設定しているはずなので，それをText1.TextとText2.Textから読み出します。移動距離を変数dist，時間を変数tiにそれぞれ格納します。速度は，移動距離を時間で割れば得られますから，distをtiで割り，結果を変数spに代入します。この変数spの値を，速度表示用のボックスのText3.Textに代入することで，結果が表示されます。

## 単位指定の拡張

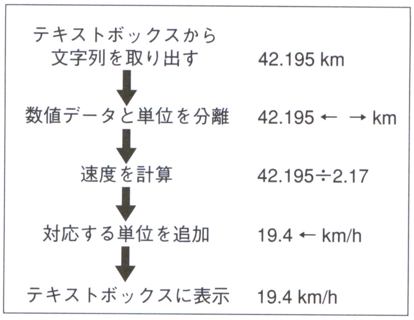
**リスト1**では，入力される距離をkm，時間を1時間単位で受け付けるようになっています。また，最終的な表示はkm単位の時速表示になっています。

これを拡張して，入力される距離や時間に付けた単位のアルファベットを識別し，的確な速度を表示するようにしてみましょう。距離の入力は「km」か「m」，時間の入力は「h」（時間）か「s」（秒）を指定することにし，入力に合わせた速度単位で結果を表示することにします。

Form_Load部分については特に変更は必要ありませんが，Command1_Click

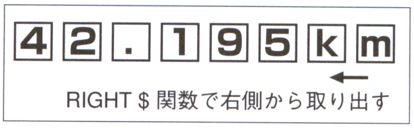
《リスト2》距離，時間の単位を識別し，速度を計算するプログラム・リスト

```
VERSION 5.00
Begin VB.Form Form1
  Caption        =  "Form1"
  ClientHeight   =  2028
  ClientLeft     =  48
  ClientTop      =  324
  ClientWidth    =  6888
  LinkTopic      =  "Form1"
  ScaleHeight    =  2028
  ScaleWidth     =  6888
  StartUpPosition =  3  'Windows の既定値
  Begin VB.CommandButton Command1
    Caption        =  "Command1"
    Height         =  492
    Left           =  2400
    TabIndex       =  5
    Top            =  1440
    Width          =  1932
  End
  Begin VB.TextBox Text3
    Height         =  372
    Left           =  4920
    TabIndex       =  2
    Text           =  "Text3"
    Top            =  600
    Width          =  1332
  End
  Begin VB.TextBox Text2
    Height         =  372
    Left           =  1920
    TabIndex       =  1
    Text           =  "Text2"
    Top            =  840
    Width          =  1452
  End
  Begin VB.TextBox Text1
    Height         =  372
    Left           =  1920
    TabIndex       =  0
    Text           =  "Text1"
    Top            =  360
    Width          =  1452
  End
  Begin VB.Label Label3
    Caption        =  "Label3"
    Height         =  372
    Left           =  3600
    TabIndex       =  6
    Top            =  600
    Width          =  1092
  End
  Begin VB.Label Label2
    Caption        =  "Label2"
    Height         =  372
    Left           =  240
    TabIndex       =  4
    Top            =  840
    Width          =  1452
  End
  Begin VB.Label Label1
    Caption        =  "Label1"
    Height         =  372
    Left           =  240
    TabIndex       =  3
    Top            =  360
    Width          =  1452
  End
End
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Private Sub Command1_Click()
   Dim dist1$, ti1$, sp1$
   Dim dist2#, ti2#, sp2#
   dist1$ = Text1.Text 'テキストボックスに入力された距離を調べる
   ti1$ = Text2.Text 'テキストボックスに入力された時間を調べる
   dist2 = Val(dist1$) '距離指定文字列を数値に変換
   ti2 = Val(ti1$) '時間指定文字列を数値に変換
   sp2 = dist2 / ti2 '速度の計算
   sp1$ = Format$(sp2, "#####.00") '計算した速度を文字列に変換
   If Right$(dist1$, 2) = "km" Then '距離の単位のチェック
      sp1$ = sp1$ + "km/"
   Else
      sp1$ = sp1$ + "m/"
   End If
   If Right$(ti1$, 1) = "h" Then '時間の単位のチェック
      sp1$ = sp1$ + "h"
   Else
      sp1$ = sp1$ + "s"
   End If
   Text3.Text = sp1$ '計算結果をテキストボックスに代入
End Sub

Private Sub Form_Load()
   Command1.Caption = "計算実行" 'キャプションの設定
   Label1.Caption = "移動距離"
   Label2.Caption = "時間"
   Label3.Caption = "速度"
   Text1.Text = "" 'テキストボックスのクリア
   Text2.Text = ""
   Text3.Text = ""
End Sub
```




**《第3図》単位指定を受け付けるための処理**

部分は入力されるデータが数値から文字列に変わるため，プログラムの変更が必要です。流れとしては**第3図**のように，まず文字列として入力されたデータの数値部分と単位部分を分離し，その後速度を計算して，最初に分離した単位部分に対応する速度単位を追加して表示してやります。




**《第4図》数値と単位の分離**

数値と単位の分離は，文字列処理関数を利用して行います。入力される文字列は**第4図**のように文字列末尾に単位が付いているはずですから，これをright $ 関数で取り出します。移動距離の場合，取り出した結果が「km」であればkm単位とみなし，他の場合はm単位とみなします。同様に，時間については「h」ならば時間単位，それ以外は秒単位とみなします。あらかじめ速度の数値を示す文字列を計算しておき，その文字列に対応する単位を追加することで表示すべき文字列を作ります。

この拡張を行ったプログラムが**リスト2**です。これを実行し，**写真3**のように単位を付けた数値を入力すると，それに

**《写真3》単位に対応させた速度計算プログラムの実行画面**

**《リスト3》速度をテーマにした「追いかけゲーム」のプログラム・リスト**

```
VERSION 5.00
Begin VB.Form Form1
   Caption         =   "Form1"
   ClientHeight    =   5550
   ClientLeft      =   60
   ClientTop       =   345
   ClientWidth     =   7500
   LinkTopic       =   "Form1"
   ScaleHeight     =   5550
   ScaleWidth      =   7500
   StartUpPosition =   3  'Windows の既定値
   Begin VB.CommandButton Command2
      Caption         =   "Command2"
      Height          =   495
      Left            =   5880
      TabIndex        =   8
      Top             =   480
      Width           =   1455
   End
   Begin VB.Timer Timer1
      Left            =   3720
      Top             =   600
   End
   Begin VB.CommandButton Command1
      Caption         =   "Command1"
      Height          =   495
      Left            =   4320
      TabIndex        =   3
      Top             =   480
      Width           =   1335
   End
   Begin VB.TextBox Text1
      Height          =   270
      Left            =   5280
      TabIndex        =   2
      Text            =   "Text1"
      Top             =   120
      Width           =   2055
   End
   Begin VB.PictureBox Picture1
      Height          =   3855
      Left            =   120
      ScaleHeight     =   3795
      ScaleWidth      =   7155
      TabIndex        =   0
      Top             =   1560
      Width           =   7215
   End
   Begin VB.Label Label5
      Caption         =   "Label5"
      Height          =   255
      Left            =   4200
      TabIndex        =   7
      Top             =   1080
      Width           =   3135
   End
   Begin VB.Label Label4
      Caption         =   "Label4"
      Height          =   255
      Left            =   4200
      TabIndex        =   6
      Top             =   120
      Width           =   975
   End
   Begin VB.Label Label3
      Caption         =   "Label3"
      Height          =   255
      Left            =   120
      TabIndex        =   5
      Top             =   1080
      Width           =   3135
   End
   Begin VB.Label Label2
      Caption         =   "Label2"
      Height          =   255
      Left            =   120
      TabIndex        =   4
      Top             =   600
      Width           =   3135
   End
   Begin VB.Label Label1
      Caption         =   "Label1"
      Height          =   255
      Left            =   120
      TabIndex        =   1
      Top             =   120
      Width           =   3135
   End
End
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Dim gs% 'ゲームシーケンス状況
Dim p1y!, p1yorg 'ターゲット位置
Dim p1v!, p1vorg 'ターゲット速度
Dim gy! 'ゴールまでの距離
```

```
Dim p2y! 'プレイヤー位置
Dim p2v! 'プレイヤー速度
Dim xsize!, ysize! 'ピクチャボックスのサイズ
Dim xscale!, yscale! '内部座標とピクチャボックスのスケーリング値
Dim t! '経過時間
Dim dt! '刻み時間

Private Sub Command1_Click()
    p2v = Val(Text1.Text) '入力された速度を取得
    p2y = 0
    p1y = p1yorg
    p1v = p1vorg
    t = 0 '時刻カウンタ
    gs = 2 'シミュレーションスタート
End Sub

Private Sub Command2_Click()
    gs = 0
End Sub

Private Sub Form_Load()
    Command1.Caption = "実行" 'キャプションの設定
    Command2.Caption = "再設定"
    Form1.Caption = "追いかけゲーム"
    Label1.Caption = ""
    Label2.Caption = ""
    Label3.Caption = ""
    Label4.Caption = "速度(m/s)"
    Label5.Caption = ""
    Text1.Text = "" 'テキストボックスのクリア
    dt = 0.01 '刻み時間
    gs = 0 'ゲームシーケンス
```

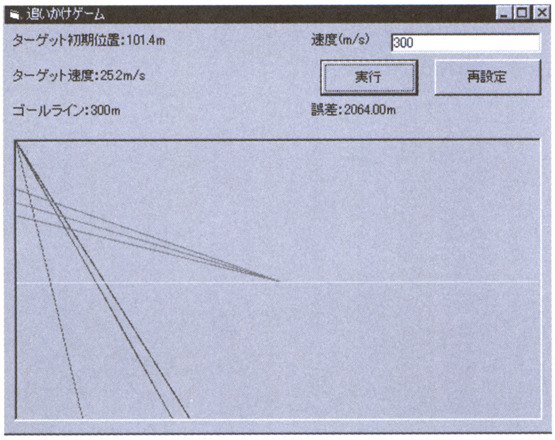
```
    Timer1.Interval = 20 'インターバルタイマ起動
End Sub

Private Sub Timer1_Timer()
    If gs = 0 Then '状況の作成
        p1yorg = Rnd(1) * 100 + 100
        p1vorg = Rnd(1) * 20 + 10
        gy = 300
        p2y = 0
        Label1.Caption = "ターゲット初期位置：" + Format$(p1yorg, "###.0m")
        Label2.Caption = "ターゲット速度：" + Format$(p1vorg, "###.0m/s")
        Label3.Caption = "ゴールライン：" + Format$(gy, "###m")
        xsize = Picture1.Width 'ピクチャボックスのサイズを取得
        ysize = Picture1.Height
        xscale = xsize / ((gy - p1yorg) / p1vorg * 2) 'X座標スケーリング値計算
        yscale = ysize / 2 / gy 'Y座標スケーリング値計算
        Picture1.Line (0, gy * yscale)-(xsize, gy * yscale), RGB(0, 255, 0)
        gs = 1
    ElseIf gs = 2 Then 'シミュレーション
        Picture1.PSet (t * xscale, p1y * yscale), RGB(255, 0, 0) '物体位置表示
        Picture1.PSet (t * xscale, p2y * yscale), RGB(0, 0, 255)
        t = t + dt '時間経過
        p1y = p1y + p1v * dt '座標更新
        p2y = p2y + p2v * dt
        If p1y > gy Then 'ターゲットがゴールラインに到達したらゲーム終了
            gs = 3
        End If
    ElseIf gs = 3 Then '結果表示
        Label5.Caption = "誤差：" + Format$(Abs(p2y - gy), "##0.00m")
    End If
End Sub
```

対応した単位が付いた速度が表示されます。また，format＄関数を利用して，速度表示の書式を指定し，見やすい表示になるようにしています。

距離を示す単位にはkmやm以外に，もっと小さいcmやmmもあります。時間も分を示すmや，dayなども考えられます。**リスト2**をこれらに対応させてみてください。

## 追いかけゲーム

それでは，速度をテーマにしたゲーム「追いかけゲーム」をプレイしてみましょう。プログラムは**リスト3**です。




**《写真4》追いかけゲーム**

ゲームでは，プレイヤーよりもある程度ゴールラインに近い場所にターゲットが存在し，ターゲットはゴールライン目指して一定速度で移動していきます。プレイヤーは，ターゲットと同時にゴールラインに到達するように，プレイヤーの速度を決めます。速度を入力した後，実際にその条件での動きをシミュレートし，ターゲットがゴールラインに到達した瞬間のプレイヤーのゴールラインとの差が最終結果となります。なるべく差が小さくなるように，速度を調整してください。

ゲームを開始すると，**写真4**のような画面が表示されます。ウィンドウ左上にはターゲットやゴールラインについての情報が表示されます。右上にはプレイヤーが速度を入力するテキストボックスと，シミュレーションを開始する実行ボタン，ターゲットの条件を変更する再設定ボタンがあります。速度を入力して実行ボタンを押すと，ウィンドウのピクチャボックスにターゲット（赤）とプレイヤー（青）の位置が，横軸時間，縦軸距離のグラフとして表示されます。

速度を適当に決めてやると，差はあまり小さくなりません。正確にゴールラインで追いつくようにするためには，条件から的確な速度を逆算する必要があります。

ポイントになるのはターゲットがゴールラインに到達するまでの時間です。プレイヤーが設定できるのはプレイヤー自身の速度だけですから，ターゲットがゴールする時間は変更できません。よって，まずはターゲットがゴールするまでの時間を計算し，その時間でプレイヤーがゴールするように速度を決めてやればよいことになります。条件から必要な速度を計算して入力してやると，差は1m以下にすることができます。

## おわりに

速度の単位は，対象となる物体によって変化します。たいていはkm/hですが，風速はm/sだったりしますし，地殻変動の速度は1年当たり数mmというオーダーになります。正確な速度表現が必要な時には，距離の単位だけでなく，時間の単位も一緒に表現しておく必要があるわけです。

次回は，速度の変化量である加速度について説明します。

21世紀の遊びはコレ！ ロボ道場

# AIBOを無線LANでコントロール！

取材／文　松原拓也

## AIBO Navigator 2（ナビゲーター）

バージョンアップされたAIBOのナビゲーター(AIBO Navigater2)に，さっそくチャレンジ！　使い心地は快適，まるでAIBOになった気分？

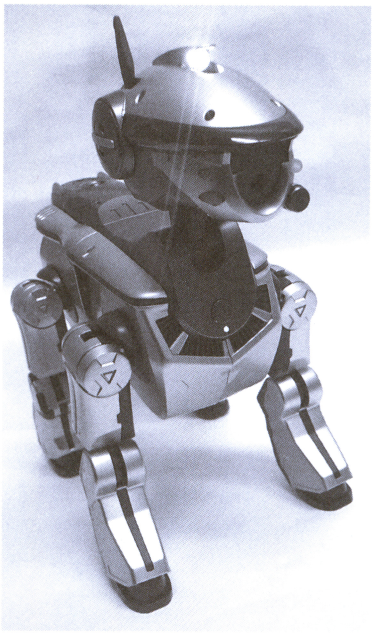
### AIBOナビゲーターとは

『AIBO ナビゲーター2』はPC画面上のコックピットからAIBOをコントロールすることができるAIBO-wareです。AIBOの視点を楽しみながら，AIBOを思いどおりに操ることができます（**写真1**）。

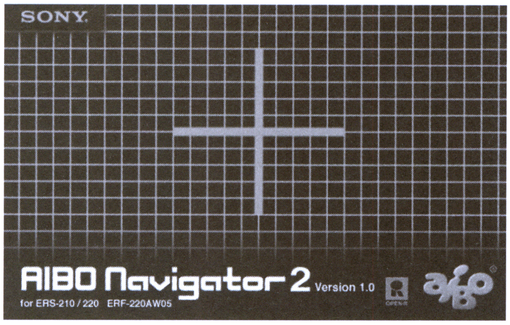
### AIBOとLAN接続

では，実際に『AIBO ナビゲーター2』を使ってAIBOを動かしてみましょう。

まず，AIBO ERS-210/220の体内にあるPCカードスロットに「ワイヤレスLANカード」を装着します（**写真2**）。それから専用のツールを使い，AIBO自身に固有のIPアドレスを割り振ります。設定したIPアドレスはAIBO専用のメモリースティック（**写真3**）に書き込まれ，同じくパソコン側にもIPアドレスを設定します。AIBOの電源を入れ，それからパソコン側のソフトを実行すると，AIBOとパソコンの接続が完了します。



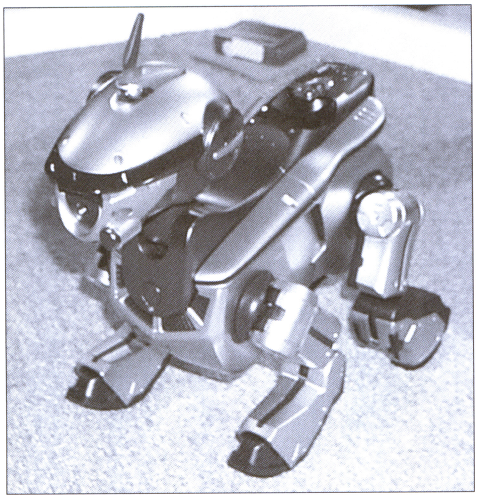
『AIBO ERS-220』
価格：180,000円（税別）／発売中
ソニー
WEBサイト
http://www.aibo.com/




『AIBO ナビゲーター2』
ERF-220AW05
（ERS-210/220用）
価格：15,000円（税別）
対応OS：Windows ME/98SE/2000
CPU：Intel Pentium III 500MHz相当以上
・無線LAN対応（IEEE802.11b準拠）のPC，又は無線LANのアクセスポイントとLAN接続されたPCが必要です。
・製品にジョイスティックは付属しません。
・AIBOワイヤレスLANカード ERA-201D1が必要です。
・有償で『AIBO ナビゲーター ERF-210AW05』からのアップグレードも行っています。
『AIBOワイヤレス LANカード　ERA-201D1』20,000円(税別)。
『AIBOナビパック ERF-220AP05』（AIBOナビゲーター2とAIBOワイヤレスLANカードのセット）29,800円（税別）。

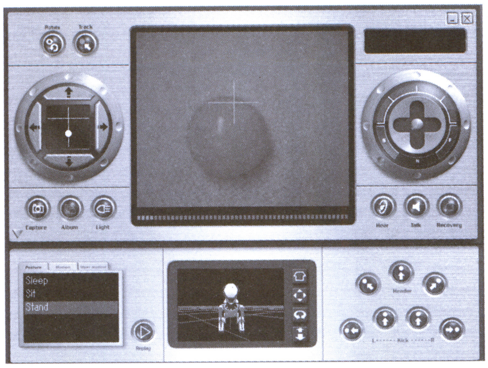
《写真1》AIBOナビゲーターの動作環境

## 動作原理



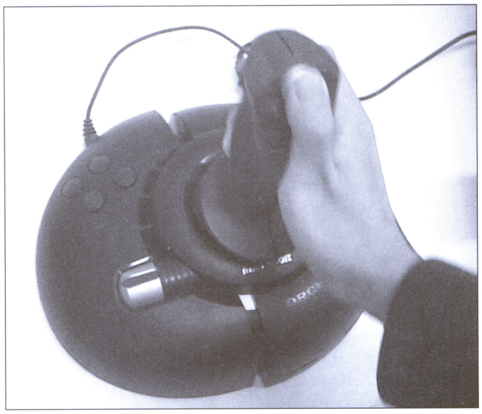
(A)

(B)




(C)

(D)



AIBOとパソコンとの通信には，「IEEE802.11b」という規格のワイヤレスLAN（無線LAN）が採用されています。「IEEE802.11b」の転送速度は最大11Mbps。通信可能範囲は室内で約20m，障害物の全くない状態で約100mと，かなり広範囲です。

『AIBO ナビゲーター 2』はフォースフィードバックに対応しています。ジョイスティックの入力情報やジョイスティックを振動させる情報はDirectInput（DirectX）を通じて行われます。

（A）AIBOが見た映像やAIBOが感じたセンサの状態をAIBOからパソコンに送られます。

（B）パソコンから，AIBOを操縦するための命令を送ります。

（C）AIBOがキックをした時や，AIBOのセンサが反応した時，その衝撃が振動となってジョイスティックに伝わります。

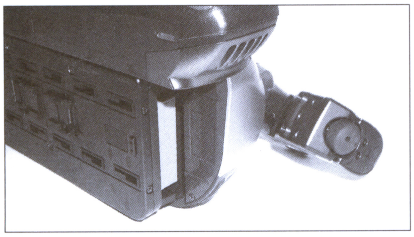
（D）ジョイスティックの入力情報がパソコンに送られます。

### AIBOの基本操作

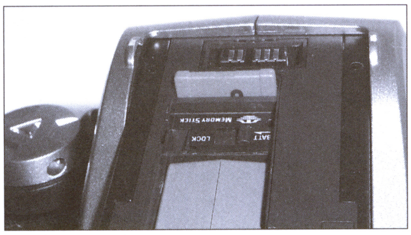
カメラの映像はパソコン上の「カメラ実況エリア」に表示されます。映像は少々荒いですがリアルタイムに表示されるので，臨場感は抜群です（写真4，5）。

AIBOをコントロールできる要素は豊富です。まず，前進，後退，右回転，左回転などの移動では，速度を3段階に調整できます。カメラアングル（首の向き）の変更は移動と独立していて，「首を横に向けたまま前進」といったことも可能です。なお，ERS-220で新たに採用された「リトラクタブルヘッドライト」の点灯にも対応しています。

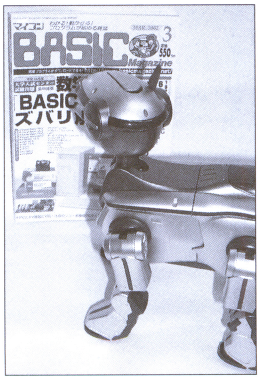
操作機器はマウスの他，ジョイスティックにも対応しています（マイクロソフト製の「サイドワインダーフォースフィードバック2」を推奨）。ジョイスティックのレバーを倒すとAIBOが歩き始めます（歩くと映像がかなり揺れます）。トリガを押すとAIBOがキックやヘディングをします（キーの対応は変更可能です）。



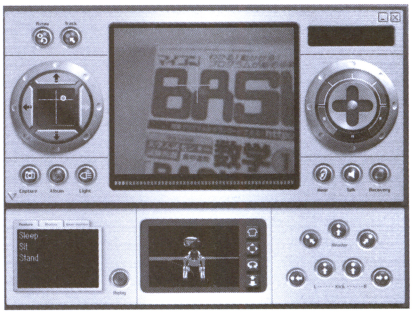
《写真2》「ワイヤレス LANカード　ERA-201D1」



《写真3》「AIBOナビゲーター」用のメモリースティック



◀《写真4》AIBOのカメラでベーマガを撮影



《写真5》写真4の状態で撮った映像。きちんと文字が読める

## カメラ機能

「スチルキャプチャーボタン」を押すと，カメラの映像を静止画として取り込むことができます。撮影した画像はアルバムウインドウで一覧表示，さらにファイルとして保存することができます。

## モーションの実行

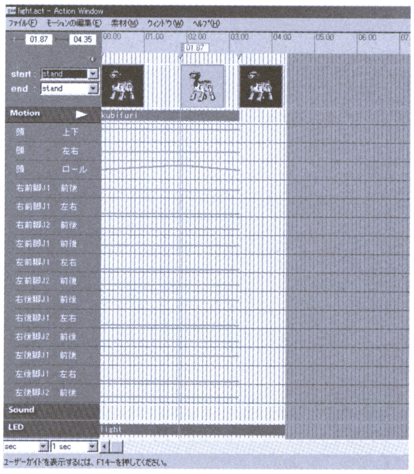
キック，ヘディング，お座り，立ち上がり，ダンスなど，AIBOの一連の動きはあらかじめ「モーション」として登録されています。このモーションにより，AIBOの複雑な動きを簡単に再現することができます。

さらに『AIBO マスタースタジオ ver.1.1』を使えば，オリジナルのモーションを作成することができます（写真6）。

## 音声の送受信

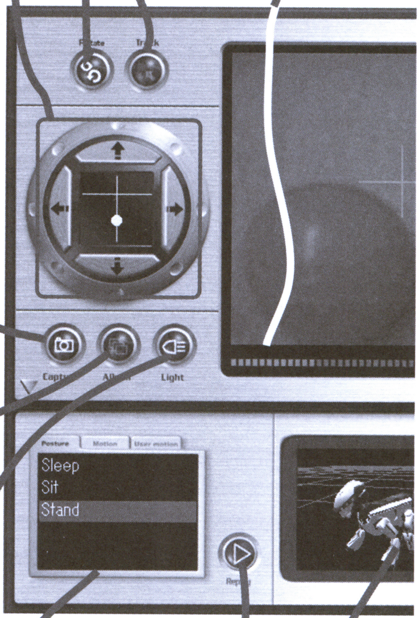
パソコン側からサウンドファイル（WAV）を指定することで，AIBOにさまざまな声や音楽を出力させることができます。

「ヒアーボタン」を押すことで，逆に音の入力も可能です。これで，AIBOの耳（マイク）がとらえた音をパソコン側で聞くことができます。この機能をうまく使って，遠く離れた人と会話することも可能ではないでしょうか（AIBOに喋らせる素材をあらかじめ用意しておかないといけませんが）。



《写真6》『AIBO マスタースタジオ ver.1.1』でモーションの作成

# 操作方法




「回転ボタン」
AIBOの体の向きをカメラアングルに合わせます。

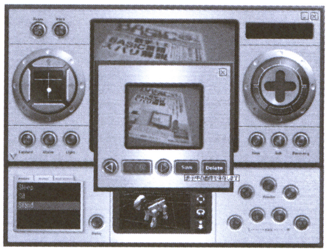
「トラッキングボタン」
ONにするとAIBOが付属のピンクボールを追尾します。

「カメラアングル表示エリア」
AIBOのカメラアングルを黄色で表示します。

「カメラアングル操作ボタン」
AIBOのカメラアングルを微調整できます。

「物体センサー」
カメラ実況エリアの，十印の位置にある物体までの距離を表示します。

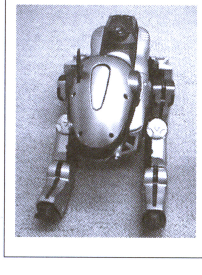
「スチルキャプチャーボタン」
AIBOのカメラに映っている画像を静止画として保存します。



「アルバムボタン」
アルバムウインドウを表示します。

「ライトボタン」
リトラクタブルヘッドライトが点灯します。

「リプレイボタン」
直前に再生したモーションを，AIBOがもう一度再生します。

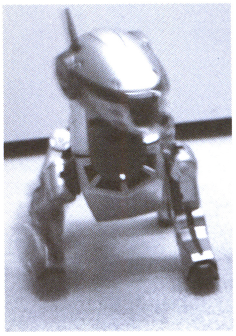


「モーション選択メニュー」
モーションを選択すると，AIBOがモーションを再生します。

「3D AIBO表示エリア」
AIBOの現在の姿勢を3D表示します。

「カメラ実況エリア」
AIBOのカメラに映っている画像を表示します。

「ステータスインジケータ」
AIBOの現在の状態を表示します。



「移動方向エリア」
AIBOの移動方向を指示します。

「ヒアーボタン」
ONにすると，AIBOがとらえた音をPCへ送信します。

「トークボタン」
PCからサウンドデータを送信し，AIBOに再生させます。

「転倒復帰ボタン」
AIBOを転倒状態から復帰させます。

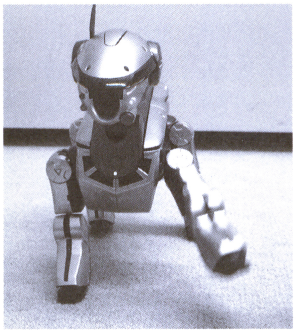
「ヘディングボタン」
AIBOがヘディングします。

「ビューホームポジションボタン」
AIBOの3D画像の視点をホームポジションに戻します。

「ビュー移動ボタン」
AIBOの3D画像の位置を動かします。

「ビュー回転ボタン」
AIBOの3D画像を回転させます。

「ビューズームボタン」
AIBOの3D画像を拡大表示します。



「キックボタン」
AIBOが前脚キックします。

## 最後に

一般にペットロボットとして親しまれているAIBOですが，この『AIBO ナビゲーター2』は，「AIBOを操作する」という全く違うアプローチを試みた画期的な製品です。

AIBOで室内を探検させてみたり，AIBOでサッカーをさせてみたりすると面白そうです。

### AIBOの歴史

1999年6月に初登場したAIBO「ERS-110」は国内向け3000体が20分で完売するという衝撃的なデビューを飾りました（価格は25万円）。

1999年11月に発売された「ERS-111」ではボディーカラーとデザイン（耳の形状，尻尾の関節）が若干修正されました。

2000年2月には受注生産体制が確立。2000年末にはデザインを“子ライオン型”に一新，いくつかの新機能が追加した「ERS-210」が発売されました（価格は15万円）。

その後，2001年9月にクマイヌ型の「ラッテ＆マカロン」と呼ばれる「ERS-311/312」が発売（価格は9万8000円）。

2001年11月には近未来的なデザインを採用した「ERS-220」が発売され，現在に至っています。

時代を先取り！ 目指せ最先端プログラマー！

# .NET（ドットネット）プログラミング入門

## 第2回 .C#言語（前編）

廣瀬 貴一

### C# と .NET

C# 言語は，Microsoft が .NET の開発用言語として発表した高度なオブジェクト指向言語です。Visual Basic の開発効率と C++ の強力さをもつという位置づけを目指して開発され，MSIL に変換されることをあらかじめ想定されていることからも，.NETに最も適した言語であることは間違いないでしょう。そして，C++ や Java 等のオブジェクト指向言語の教訓を踏まえているため，これまで存在していた C++ やJava の短所をカバーし，XML などの Web との技術の連帯も考えられた注目の言語でもあります。

この場では，C++ や Java にはない C# の新しい言語仕様などをピックアップして紹介したいと思います。

### C# はどこで手に入れる？

C# 言語は .NET 開発環境に含まれています。.NET アプリケーションの開発には，.NET Framework SDKと呼ばれる開発環境を用います。これは，Microsoft の **http://www.microsoft.com/japan/developer/net/dotnetsdkdown.asp** からダウンロードすることが可能です。 .NET に関連する情報や開発環境の仕様など，細かい情報もMicrosoftのサイトで確認することができるので，定期的にチェックすると良いでしょう。また， Visual Studio .NET のベータ版でも開発することができます。.NET Framework SDK は GUIでの開発をサポートしないので，コンソールの知識がない方は VS .NET を使用してください。

ただし，この場では .NET Framework SDK を使用していることを想定して話を進めます。

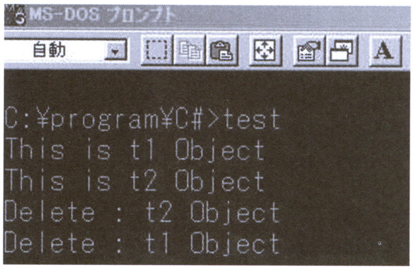
### クラスの宣言と利用

**クラスの宣言は従来の言語同様に class キーワードを用います。**

*[attributes] [modifiers] class identifier [:base-list] { class-body }[;]*
*attributes:属性群*
*modifiers:クラス修飾子群*
*identifier:識別子群*
*base-list:クラスベース*
*class-body:クラス本文*

**《リスト1》インスタンスの生成と初期化**

```
class Test {
		public string str;
		public Test(string str) {
				this.str = str;
		}
		~Test() {
				System.Console.WriteLine("Delete : " + str);
		}
		static void Main() {
				Test t1 = new Test("t1 Object");
				Test t2 = new Test("t2 Object");
				System.Console.WriteLine("This is " + t1.str);
				System.Console.WriteLine("This is " + t2.str);
		}
}
```




**《写真1》リスト1の実行画面**

メンバの宣言は C++ と異なり，各メンバにアクセス可能性(public，protected，internal，protected internal，private)を指定する必要があります。クラスメンバ宣言に修飾子が含まれていない場合，宣言のデフォルトは private となります。宣言したクラスのインスタンスを作成するには new 演算子を用います。 new 演算子は作成したインスタンスの参照を返します。参照を受けるのは，そのクラスに互換性のある型でなければなりません（**リスト1**）。

Main() メソッドで new 演算子を用いコンストラクタを指定しています（**写真1**）。Test() コンストラクタは適切な初期化処理をした後，インスタンスの参照を返します。

注意してほしいのは ~Test() デストラクタです。C++と異なり，インスタンスの破棄のタイミングはCLRが決定します。

プログラムが明示的にデストラクタを呼び出したり，インスタンスを破棄することはありません。インスタンスは，コードがインスタンスを使用する可能性がなくなるとデストラクションの対象となり，インスタンスがデストラクションの対象になれば，その後何らかのタイミングでデストラクタが呼び出されます。

## プロパティ

C++ にはなかった強力な機能の一つに，プロパティがあります。C# 言語は，クラス利用者が内部仕様を知らなくても，できる限り直感的に扱えるように設計されており，このプロパティという概念も，直感的なソースを実現するための機能です。プロパティは，フィールドにアクセスされる時，暗黙的にアクセッサメソッドを呼び出すサービスです。クラスを利用する開発者は，フィールドに直接アクセスする感覚でアクセッサメソッドを呼び出せます。

*[attributes] [modifiers] type identifier {accessor-declaration}*
*attributes:属性群*
*modifiers:プロパティ修飾子群*
*type:型*
*identifier:[インターフェイス型.]識別子*
*accessor-declaration:アクセッサ宣言*

これがプロパティの宣言です。クラス利用者は，プロパティの識別子を指定するだけで暗黙的にアクセッサを呼び出すことができるのです。アクセッサ宣言は，次のようになります。

*set {accessor-body}*
*get {accessor-body}*
*accessor-body:アクセッサ本文*

set アクセッサは，プロパティに対して何らかの値が代入された時に呼び出され，逆に get アクセッサは，プロパティが式で用いられた時に呼び出されます。set アクセッサは，プロパティ型の value という暗黙的な変数をもち，この変数にはプロパティに代入された値が格納されています。

《リスト2》プロパティとアクセッサ宣言

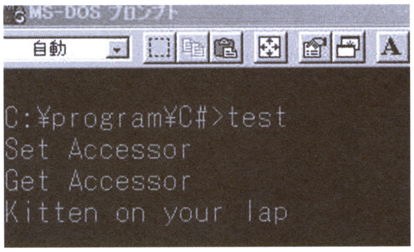
```
class Property {
        private string str;
        public string Str {
                get {
                        System.Console.WriteLine("Get Accessor");
                        return str;
                }
                set {
                        System.Console.WriteLine("Set Accessor");
                        str = value;
                }
        }
}

class Test {
        static void Main() {
                Property obj = new Property();
                obj.Str = "Kitten on your lap";
                System.Console.WriteLine(obj.Str);
        }
}
```

getアクセッサに暗黙的な変数はなく，プロパティ型の戻り値をもっています。すなわち，set アクセッサはプロパティ型の value という引数をもち void を返すメソッドに等しく，getアクセッサは引数を受け取らず，プロパティ型の戻り値をもつメソッドに等しいと考えられるのです（**リスト2**）。

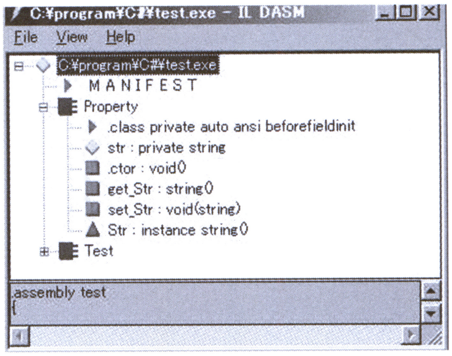
もし，呼び出し専用のフィールドであれば getアクセッサのみを指定することで，呼び出し専用のプロパティを作成することも可能です。これによって，プログラマーは必要な情報にアクセスする時，従来のような get...() や set...() といったメソッドを覚えたり，リファレンスを参照する手間が省けます。

プロパティのアクセッサ宣言は，MSIL にコンパイルされる時，アクセッサメソッドに変換されます。この時，メソッドの名前は"set_プロパティ名" や "get_プロパティ名"というようになります。（**写真2**）

ILDASM で解析した，先ほどのプログラムの MSIL です。Str プロパティのアクセッサがメソッド に変換されていることがわかります（**写真3**）。



《写真2》リスト2の実行画面



《写真3》リスト2のMSIL

## インデクサ

インデクサは，技術的にプロパティと類似したもので，暗黙的にメソッドを呼び出すサービスの一つです。インデクサを用いれば，オブジェクトを配列のように扱うことができるようになります。これは，オブジェクトに添字を指定することで，適切なメソッドを呼び出すように設計されています。プロパティと違うのは，インデクサは識別子ではなく，シグネチャでメソッドを検索します。

*[attributes] [modifiers] indexer-declarator {accessor-declarations}*
*attributes:属性群*
*modifiers:インデクサ修飾子群*
*indexer-declarator:インデクサ宣言子*
*accessor-declarations:アクセッサ宣言子群*

アクセッサ宣言子群に指定するのは，プロパティ同様に set と get アクセッサ

です。インデクサアクセスは，クラス，構造体，インターフェースの参照，または値の要素アクセスです。要素に対して代入すればsetアクセッサが，また式中に用いられればgetアクセッサが呼び出されます。インデクサ宣言子は次のいずれかで宣言します

*type this [formal-index-parameter-list]*
*type interface-type.this [formal-index-parameter-list]*
*type:型*
*interface-type:インターフェイス型*
*formal-index-parameter-list:仮パレメータリスト*

インデクサはシグネチャで判断されるため，オーバーロードすることが可能です。アクセッサは，仮パラメータリストで指定された値を受け取ります。さらに，setアクセッサは，代入された値をインデクサ型の暗黙的な変数valueとして受け取ることができ，getアクセッサは，インデクサ型の戻り値をもっています。

**リスト3**は，論理パレットをシュミレートするPaletteクラスにインデクサを採用したものです。プログラムを実行すると，標準出力にグレーシェードのカラー配列を表す値が表示されます。パレットは256色のエントリを表現するために，int型のカラー情報を格納するフィールドをもちます。

このフィールドは直接公開せず，インデクサによって間接的にアクセスするとより，直感的に扱えます。インデクサによってMain()メソッドからは，pal[count]というように要素アクセスによる直感的な利用が可能になっています。

インデクサの良い代表例は文字列を表すSystem.Stringクラスで用いられています。アクセッサの内部でindexを調べて，不正な要素アクセスがないかを調べれば，より安全なプログラムを作ることができます。

**《リスト3》インテグサを用いたバレットクラス**

```
class Palette {
		int[] palColor = new int[256];
		public int this [int index] {
				set { palColor[index] = value; }
				get { return palColor[index];  }
		}
}

class Test {
		static void Main() {
				Palette pal = new Palette();
				for(int count = 0 ; count < 65 ; count++)
						pal[count] = ((4 * count) << 16) + ((4 * count) << 8) + 4 * count;
				pal[64] = 0xFFFFFF;

				for(int count = 0 ; count < 65 ; count++)
						System.Console.WriteLine(pal[count].ToString("X6"));
		}
}
```

## オーバーライド

C#の継承は，C++と異なり多重継承を認めません。宣言するクラスがもてる基底クラスは常に一つです。何も継承しない場合，そのクラスの基底クラスはSystem.Objectクラスになります。C#言語では，あらゆるクラスがSystem.Objectを直接，または間接的に基底クラスとしています。

これによって，.NETは一貫した型統一システムを提供しているのです。また，多重継承ができない代わりにインターフェイスを実装することで，それぞれのクラスは共通の操作特性を保証することができます。

基底クラスのメンバと名前が衝突した場合は，デフォルトで基底クラスのメンバが隠蔽されます。この場合，一般的にコンパイラは開発者が知らずに基底クラスのメンバを隠蔽した可能性を考慮して，警告を発生させます。基底クラスのメンバを隠蔽する場合，それを明示するためにnew修飾子を用います。

**《リスト4》基底クラスのメンバの隠蔽**

```
class Base {
		public void M() {
				System.Console.WriteLine("This Method is Base.M()");
		}
}

class Test : Base {
		new public void M() {
				System.Console.WriteLine("This Method is Test.M()");
		}
		static void Main() {
				Base obj = new Test();
				obj.M();
		}
}
```

**リスト4**では，Baseクラスとそれを継承するTestクラスでM()メソッドが衝突しています。Testクラスではnew修飾子を用いて隠蔽を明示しています。Main()メソッドでは，Testのインスタンスを作成し，その参照をBase型の変数に代入しています。コンパイラは，インスタンスの実体に関係なく参照型で実行するべきメソッドを決定するためBase.M()が実行されることになります。

Base obj = new Test(); の文をTest obj = new Test() に変更すれば，Test.M()が実行されることを確認できます。このように，コンパイル時に参照型で呼び出すべきメソッドを決定する方法をアーリーバインディングと呼びます。

しかし，C#言語も当然オーバーライドをサポートしています。まず，基底クラスは，オーバーライドされるべきメソッドを**virtual**修飾子で仮想化しなければなりません。そして，それを継承したクラスで仮想メソッドと名前が衝突した場合，new修飾子で隠蔽するか**override**修飾子でオーバーライドするかを指定することができます。オーバーライドによる，実行時まで呼び出すべきメソッドを決定しない方法をレイトバインディングと呼びます。

**リスト5**を見てください。BaseクラスのM()メソッドがvirtual修飾子によって仮想化されています。virtualはコンパイラに，メソッドの実体は実行時までわからないということを通知します。Testクラスではoverride修飾子によって，M()メソッドをオーバーライドすることを示しています。

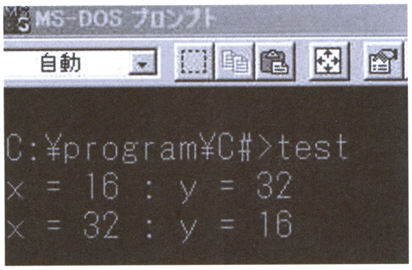
これによって，Main()メソッドのobj.M()では，実行時にobjが参照するインスタンスの実体によって呼び出すメ

《リスト5》仮想化とオーバライド

```
class Base {
        virtual public void M() {
                System.Console.WriteLine("This Method is Base.M()");
        }
}

class Test : Base {
        override public void M() {
                System.Console.WriteLine("This Method is Test.M()");
        }
        static void Main() {
                Base obj = new Test();
                obj.M();
        }
}
```

《リスト6》値型の参照渡し

```
class Test {
        static void Main() {
                int x = 16 , y = 32;
                System.Console.WriteLine("x = " + x + " : y = " + y);
                M(ref x ,ref y);
                System.Console.WriteLine("x = " + x + " : y = " + y);
        }
        static void M(ref int x , ref int y) {
                int tmp = y;
                y = x;
                x = tmp;
        }
}
```




《写真4》リスト6の実行画面

ソッドが決定されます。この変数の参照型は Base になっていますが，この実体は Test 型です。このプログラムを実行すると， Test.M() が呼び出されることを確認できるでしょう。

## 参照を渡す

C/C++ 言語では，関数の引数にポインタを渡すことで，呼び出し元の変数の内容を，他の関数から変更することができました。 C# 言語にはポインタはありませんが，値型の参照を他のメソッドに渡す方法を用意しています。メソッドが値型の参照を受け取るには，メソッドを宣言する時に **ref** メソッドパラメータ，もしくは out メソッドパラメータを指定します。

ref は，メソッドに渡される変数が確実に初期化されていなければなりません。out の場合は，初期化されていなくてもかまわない場合に使います。また，メソッドを呼び出す時も，パラメータで変数の前に ref または out を指定する必要があります。

int(System.Int32)型は値型です。つまり，普通にメソッドに渡した場合はその値のコピーが渡されるため，メソッド内でその値を変更しても，呼び出し元の変数に影響はありません。ところが，**リスト6**のプログラムのM() メソッドの場合は ref によって値型の参照を渡しているため，M() メソッド内で x や y の値を変更すると，その呼び出し元の変数にも影響が出ます。 Main() メソッドで，M() メソッドを呼び出したあと，x と y 変数の内容が入れ替わっているのがわかります（**写真4**）。

## サンプルプログラム

今回は，Windows 3.0 の DIB ビットマップを操作するアプリケーションを C# で作ってみましょう。ただし，.NET のグラフィックス API や，プラットフォームの API には一切頼らずに， C# の基本仕様と System.IO 空間の API のみを使ってビットマップファイルを制御します。

前回のサンプルでは文字データの入出力を行いましたが，今回はバイナリデータの入出力を行っていることに注目してください。CUI プログラムはグラフィカルではないので面白くないと思っている方もいるかもしれませんが，CUIでも，画像ファイルをバイナリレベルで操作することで，グラフィックツールを作れます。

**リスト7**は，コマンドプロンプトから元

《リスト7》バイナリ操作によるビットマップの反転

```
using System.IO;
using System;

class BITMAPFILEHEADER {
        public short        bfType;
        public int bfSize;
        public short        bfReserved1;
        public short        bfReserved2;
        public int bfOffBits;
        public BITMAPFILEHEADER(BinaryReader br) {
                bfType = br.ReadInt16();
                bfSize = br.ReadInt32();
                bfReserved1 = br.ReadInt16();
                bfReserved2 = br.ReadInt16();
                bfOffBits = br.ReadInt32();
        }
        public void WriteFile(BinaryWriter bw) {
                bw.Write(bfType);
                bw.Write(bfSize);
                bw.Write(bfReserved1);
                bw.Write(bfReserved2);
                bw.Write(bfOffBits);
        }
}

class BITMAPINFOHEADER {
        public int biSize;
        public int biWidth;
        public int biHeight;
        public short        biPlanes;
        public short        biBitCount;
        public int biCompression;
        public int biSizeImage;
        public int biXPelsPerMeter;
        public int biYPelsPerMeter;
        public int biClrUsed;
```

つぎのページにつづく

```
		public int biClrImportant;
		public BITMAPINFOHEADER(BinaryReader br) {
			biSize = br.ReadInt32();
			biWidth = br.ReadInt32();
			biHeight = br.ReadInt32();
			biPlanes = br.ReadInt16();
			biBitCount = br.ReadInt16();
			biCompression = br.ReadInt32();
			biSizeImage = br.ReadInt32();
			biXPelsPerMeter = br.ReadInt32();
			biYPelsPerMeter = br.ReadInt32();
			biClrUsed = br.ReadInt32();
			biClrImportant = br.ReadInt32();
		}
		public void WriteFile(BinaryWriter bw) {
			bw.Write(biSize);
			bw.Write(biWidth);
			bw.Write(biHeight);
			bw.Write(biPlanes);
			bw.Write(biBitCount);
			bw.Write(biCompression);
			bw.Write(biSizeImage);
			bw.Write(biXPelsPerMeter);
			bw.Write(biYPelsPerMeter);
			bw.Write(biClrUsed);
			bw.Write(biClrImportant);
		}
}

class NotBmp {
		public static int Main(string[] args) {
			if (args == null || args.Length < 2) {
				Console.WriteLine("notbmp <入力ビットマップファイル> <出力ビットマップファイル>");

				return 1;
			}
			BITMAPFILEHEADER bmpFileHeader;
			BITMAPINFOHEADER bmpInfoHeader;
			byte[] bPixelBits;

			FileStream fs = new FileStream(args[0] , FileMode.Open);
			BinaryReader br = new BinaryReader(fs);

			bmpFileHeader = new BITMAPFILEHEADER(br);
			if (bmpFileHeader.bfType != 0x4D42) {
				Console.WriteLine("This file is not BITMAP file");
				return 1;
			}
			bmpInfoHeader = new BITMAPINFOHEADER(br);
			if (bmpInfoHeader.biSize != 0x28) {
				Console.WriteLine("This file is not Windows3.0 BITMAP");
				return 1;
			}
			else if (bmpInfoHeader.biBitCount != 0x18) {
				Console.WriteLine("This file is not 24 bit color");
				return 1;
			}

			bPixelBits = new byte[bmpFileHeader.bfSize - bmpFileHeader.bfOffBits];
			for (int count = 0 ; count < bPixelBits.Length ; count++)
				bPixelBits[count] = br.ReadByte();

			br.Close();
			fs.Close();

			fs = new FileStream(args[1] , FileMode.Create);
			BinaryWriter bw = new BinaryWriter(fs);
			bmpFileHeader.WriteFile(bw);
			bmpInfoHeader.WriteFile(bw);

			for(int count = 0 ; count < bPixelBits.Length ; count++) {
				int tmp = (bPixelBits[count++] << 16) + (bPixelBits[count++] << 8) + bPixelBits[count];
				tmp = ~tmp;
				bw.Write((byte)tmp);
				bw.Write((byte)(tmp >> 8));
				bw.Write((byte)(tmp >> 16));
			}

			bw.Close();
			fs.Close();
			return 0;
		}
}
```

のビットマップファイル名と，出力ビットマップ名を引数で入力して起動します。

《写真５》 処理前

《写真６》 処理後

入力方法 (実行ファイル名はnotbmp.exe とする)

**notbmp <入力ビットマップファイル名> <出力ビットマップファイル名>**

このプログラムは元のビットマップ（**写真5**）の色を反転した新しいビットマップ（**写真6**）を作成します。このビットマップは単純に色を反転させただけなので，出力したビットマップを再び反転させれば元のビットマップに復元されます。

ただし，このプログラムは，OS/2 スタイルの DIB や BITMAPV4HEADER 以降の情報ヘッダをもつ DIB，カラービット数が 24 以外のビットマップを処理することはできません。

Main() メソッドでは，DIB の情報ヘッダなどを読み取り，正しいビットマップファイルかどうかを調べます。その後，ピクセルビットを bPixelBits に読み取り，これを反転させてファイルに書き込んでいます。

このプログラムの bPixelBits の処理を改造すれば，画像のフィルタリングなどを行うことができます。ぜひ，挑戦してみてください。

自作一代記

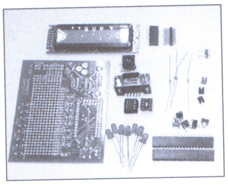
# 電子工作にチャレンジ!

連載第13回

マイコンでBASICインタプリタが動く

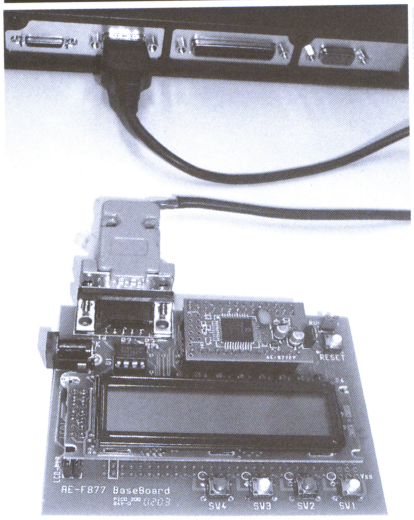
# 「AKI-PIC877モジュール用ベースボード・キット」（その2）

製作・文／松原拓也



『AKI-PIC877モジュール用ベースボード・キット』（16文字×2行液晶モジュール，シリアルポートx1，タクトスイッチ x5，I2C EEPROM，LEDx8，AKI-PIC877モジュール x1 付属） 4,600円 *
『AKI-PIC877モジュール』（8bitマイコン「PIC16F877-20pt」搭載，動作クロック20MHz，A/D機能内蔵，8KBフラッシュROM内蔵，256ByteEEPROM内蔵，RS232CドライバIC，5Vレギュレータ実装済）（単品の価格）1,700円*
*価格はPIC BASICが付属しないタイプのものです。
*PIC BASICを利用するためにはBASICインタプリタが書き込まれたAKI-PIC877モジュールが必要です。
http://www.akizuki.ne.jp/ （株）秋月電子通商

今回はなんとPICでBASICのプログラミングに挑戦します！



## 概要

「PIC BASIC」はPICで動作するBASIC言語の開発環境です。機能は四則演算，条件判断，変数の代入（変数はByte/Word/Long型。配列が使えます），I/O制御，A/D入力，LCD表示，EEPROMの読み書きなど，盛り沢山です。

## 動作原理

ベースボードにはPIC16F877というマイコンが搭載されています。マイコンにはフラッシュROMが内蔵されていて「BASICインタプリタ」と「ユーザープログラム」に分けて使われます。BASICで書かれたプログラムは中間コードの状態で，ユーザープログラム領域に格納されます（最大4KB）。実行のさい，この中間コードはBASICインタプリタを介して実行されます。

## 開発環境の構築

以下の手順で「PIC BASIC」の開発環境を構築します。

①付属のCD-ROMを使って，パソコンに「PIC BASIC」の開発ツールをインストールします。対応OSはWindowsです。

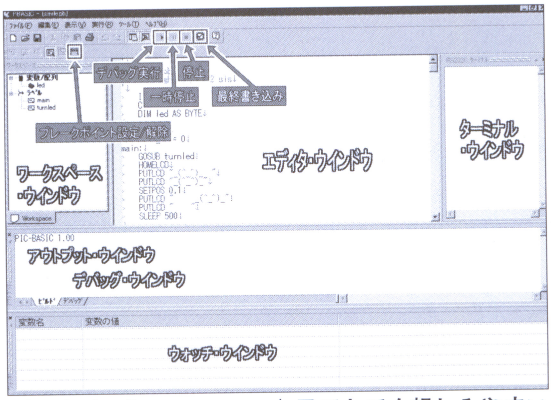
②ベースボードとパソコンをシリアルケーブル（Dsub9pinのストレート）で接続します。

③スタートメニューから「PIC BASIC」の開発ツールを起動します（**写真1**）。

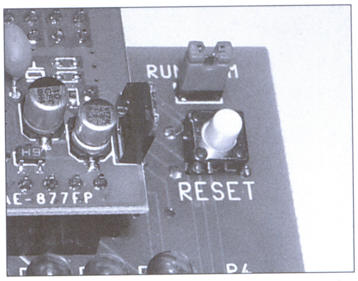
④「ツール-オプション」をメニュー選択して，シリアルのポート番号を設定します（**写真2**）。

⑤ベースボードのジャンパJ5をプログラムモードにします（**写真3**）。

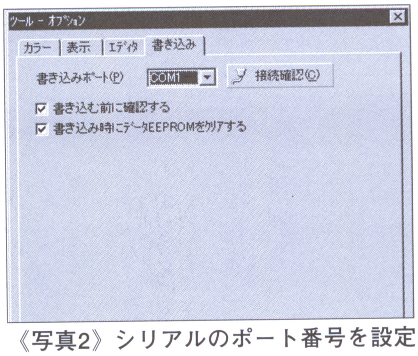
⑥DCジャックにACアダプタを接続します（**写真4**）。



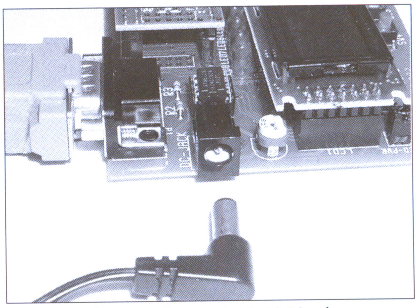

《写真1》「Visual Basic」風でとても親しみやすい「PIC BASIC」の開発ツール




《写真3》ジャンパをショートさせてプログラムモードに設定




《写真2》シリアルのポート番号を設定する



《写真4》DCジャックを接続（2.1ϕ，電圧は7〜12V，内側がプラス）

## コーディング

「エディタ・ウインドウ」にプログラムを記述します。今回はテストとしてサンプルを実行してみます。「ファイル-開く」をメニュー選択して，インストール先の“Program Files¥PIC-BASIC¥Samples”にある“smile.pb”を開きます（**リスト1**）。

この時，変数や関数名が「ワークスペース・ウインドウ」に表示されます。

## デバッグ

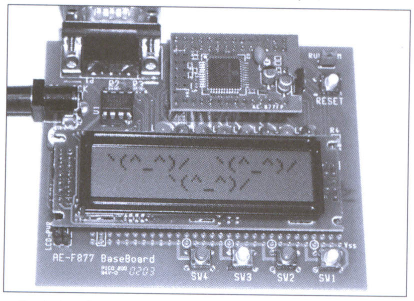
ツールバーの「実行」ボタンを押すとプログラム実行です。この時，「アウトプット・ウインドウ」には，コードのサイズ，エラー情報が表示され，プログラムがフラッシュROMに転送されます。フラッシュROMの書き込み寿命は最低1,000回です（実際はそれ以上書き込めます）。

実行すると，液晶パネルに顔文字が表示され，LEDが点灯しました（**写真5**）。ここで「一時停止」ボタンを押すと，プログラムが一時的に停止します。「実行」ボタンを押すと，停止した場所から実行を再開します。「停止」ボタンでプログラムは停止します。

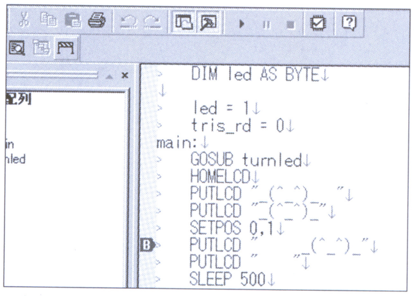
「ブレークポイントの設定／解除」ボタンで，ブレークポイントを設定できます（**写真6**）。実行中，その位置を通過するとプログラムが停止しますので，動作の確認に重宝します（**写真7**）。

「ウォッチ・ウインドウ」に変数名を登録すると，一時停止した時にその変数の値を見ることができます（**写真8**）。

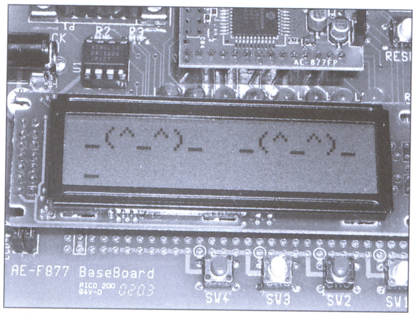
プログラム内に「DEBUG」命令を組み込んでおくと，変数や処理の様子が「デバッグ・ウインドウ」に表示されます（**写真9**）。



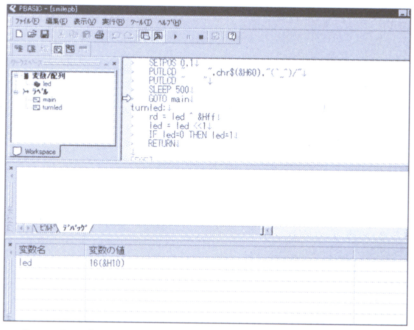

《写真5》サンプルプログラム“smile.pb”の実行結果



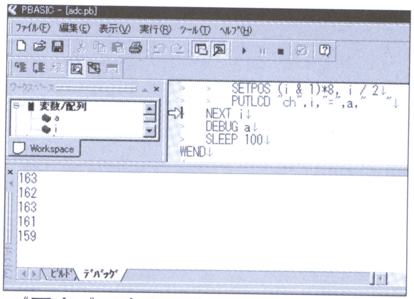

《写真6》ブレークポイントの設定（Bのマークが停止位置）



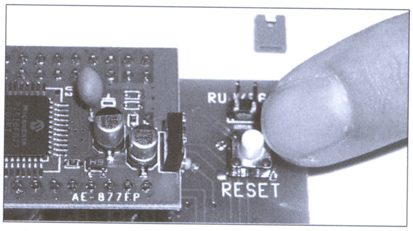

《写真7》プログラムを一時停止した時の様子



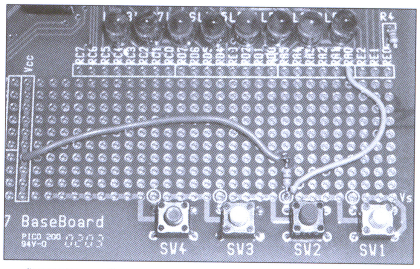

《写真8》変数の値をウォッチ・ウインドウで確認できる




《写真9》デバッグ・ウインドウにデバッグ情報を表示（更新はリアルタイム）



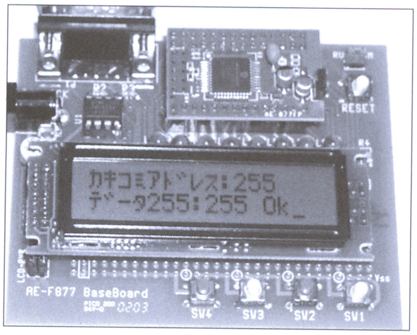

《写真10》ジャンパをオープンにしてRUNモードに設定



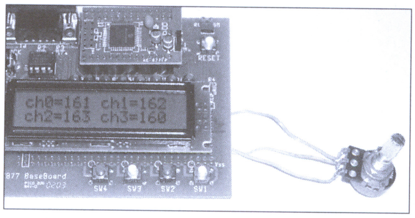

《写真11》スイッチ増設の例。抵抗でポートをプルアップ。その先端をスイッチに接続して完成

## サンプル

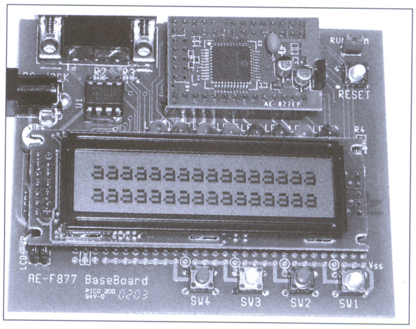
「PIC BASIC」に付属するサンプルの一例です。




▲“eeprom.pb”　内蔵EEPROM（256Byte）の読み書きをテストします（ボード上には他に1MbitのEEPROMが増設されています）。




▲“adc.pb”　A/D入力を行います（写真は可変抵抗を接続した例）。




▲“serin.pb”　シリアルポートからの通信データを液晶に表示します（写真はターミナル・ウインドウからキーボード入力を行った結果）。

## 最終書き込み

ツールバーの「最終書き込み」ボタンを押すと，デバッグ用の処理を省いたプログラムがフラッシュROMに書き込まれます。あとは，ジャンパJ5をRUNモードにしてリセット，または電源を入れ直します（**写真10**）。これで，ベースボードとパソコンの接続を切り離せる状態（スタンドアロン）になります。

## カスタマイズ

ベースボードにはユニバーサルエリアという，基板の領域があります。未使用ポート（RA0～7，RC0～2，RC5～7，RE0～7）を生かして自作することができます（**写真11**）。

## PIC界の革命！

PICライタなしでROMが焼けて，しかもICEなしでデバッグできる。今までデバッグというと，マイコン抜いて，ライタに差して，プログラム焼いて……という手間を強いられてきたわけですが，それを考えると夢のようです。

「PIC BASIC」は，扱いが簡単で，さらにライセンスフリー（ロイヤリティはPIC-BASICモジュールの価格に含まれます）。今後，業務に趣味にと，市場を席捲していきそうです。本誌にも「PIC BASIC」のプログラムが投稿されると面白いかもしれません（投稿，待ってます）。

《リスト1》サンプルプログラム“smile.pb”

```
'
'           smile.pb
'           Copyright (c) 2002 sis
'
            INITLCD
            CLEARLCD
            DIM led AS BYTE

            led = 1
            tris_rd = 0
main:
            GOSUB turnled
            HOMELCD
            PUTLCD "_(^_^)_ "
            PUTLCD "_(^_^)_"
            SETPOS 0,1
            PUTLCD "      _(^_^)_"
            PUTLCD "    "
            SLEEP 500

            GOSUB turnled
            HOMELCD
            PUTLCD chr$(&H60),"(^_^)/  "
            PUTLCD chr$(&H60),"(^_^)/"
            SETPOS 0,1
            PUTLCD "      ",chr$(&H60),"(^_^)/"
            PUTLCD "    "
            SLEEP 500
            GOTO main
turnled:
            rd = led ^ &Hff
            led = led <<1
            IF led=0 THEN led=1
            RETURN
```

「PIC-BASIC Ver.1.0命令リファレンス」

◆基本動作

| | |
|---|---|
| DIM | 変数の定義 |
| LET | 変数・配列への数値の代入（省略可） |
| GOTO | 行のジャンプ |
| GOSUB...RETURN | サブルーチンの呼出/リターン |

◆ループ・条件分岐

| | |
|---|---|
| FOR...NEXT | FOR文 |
| WHILE...WEND | WHILE文 |
| DO...UNTIL | DO文 |
| IF...THEN | IF文 |

◆I/O制御

| | |
|---|---|
| OUTPUT | ピンを出力に変更 |
| INPUT | ピンを入力に変更 |
| HIGH | ピンをハイ(H)にする |
| LOW | ピンをロー(L)にする |
| TOGGLE | ピン出力の入れ替え(H)⇔(L) |
| REVERSE | 入力⇔出力ピンの入れ替え |

◆A/D変換

| | |
|---|---|
| ADC | アナログデータ取り込み |

◆シリアル

| | |
|---|---|
| SERIN | RS232Cデータの受信 |
| SEROUT | RS232Cデータの送信 |
| SERCLEAR | RS232C受信バッファのクリア |

◆液晶モジュール

| | |
|---|---|
| INITLCD | 液晶モジュールの初期化 |
| CLEARLCD | 液晶モジュールの表示クリア |
| PUTLCD | 液晶モジュールへ文字・数値の表示 |
| SETPOS | 液晶モジュールのカーソル位置変更 |
| HOMELCD | 液晶モジュールのカーソルをホーム位置に移動 |

◆スリープ・ウェイト

| | |
|---|---|
| SLEEP | 指定時間の待機（ウェイト） |
| END | プログラムの終了 |

◆EEPROM

| | |
|---|---|
| READ | 内蔵EEPROMの読み出し |
| WRITE | 内蔵EEPROMへの書き込み |

◆I2C通信

| | |
|---|---|
| I2CREAD | I2C EEPROMの読み出し |
| I2CWRITE | I2C EEPROMへの書き込み |

◆レジスタ制御

| | |
|---|---|
| POKE | ファイルレジスタ書き込み |

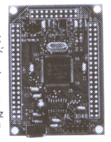
◆デバッグサポート

| | |
|---|---|
| DEBUG | デバッグ命令 |

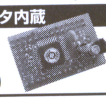

AKI-PIC プログラマーキット Ver.3
シミュレータソフト付のすぐれもの！
最新PICフラッシュマイコン版、ほぼ全てに対応！
PIC16F84 マイコンモジュールが直接焼けます！
●PIC16C5X ●PIC16F83/84 ●PIC17CX ●PIC12C508（509）
●PIC16F874・877 ●PIC16C6X対応 ●Windows98対応CD-ROMソフト付
●マイクロチップ社CD-ROMソフト付 1キット¥6,700
★マイクロチップ社のPIC(ピック)マイコンのほとんど全てがプログラマできます★28ピンゼロプレッシャソケット・40ピンゼロプレッシャソケット付★RS232C(ADM232AAN)パソコン通信回路内蔵★当社オリジナル日本語対応アセンブラ、CD-ROM付★ウィンドウズ98版ライターコントロールソフト付(シミュレータソフト付)★専用ガラスエポキシ基板使用★マイクロチップ社製品カタログCD-ROM付★テスト用PIC12C509、PIC16F84-10/P(10MHzレゾネータ付)付
専用電源セット¥1,000 15V 0.8A ACアダプタ＋9P Dサブコード1m

H8/3664モジュール開発セット
H8/3664にQFP版超小型マイコンモジュール基板登場。
★39×27mmに全機能を凝縮、さらにADコンバータは8CHに増えました★RS232用ADM232、5Vレギュレータシステムクロック発振器（16MHz）、サブ水晶発振子（32kHz）書込み制御回路等搭載済み★チップ部品の大幅採用により、外部接続用コネクタ以外は全て実装ハンダ付け済です（コネクタ付属） 1セット¥4,500
H8/3664モジュールにソフト（Cコンパイラ、アセンブラ、ライター、モニターデバッカ）、パソコン接続用ベース基板パーツ一式が付属したセット
◎2台目以降の方にH8/3664マイコンモジュールのみ 1台¥1,600

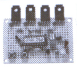
AKI-H8/3048F（フラッシュメモリ内蔵）超高速動作16MHz版（20MHz可）超高性能マイコンボード
★H8/3048F（日立）を使った超高速・超高集積ボードです ★100ピンH8CPUとRS232C ICは基板実装済みで高信頼 ★外形寸法はAKI-80と同一サイズ5×7cm ★ガラスエポキシ・スルーホール加工ボードです！
★ROM:128KB（フラッシュメモリ）★RAM:4KB（SRAM）★多機能タイマ:16bit×5ch、PWM:5相MAX他 ★シリアル:2ch（同期、非同期切替え可、ボーレートジェネレータ内蔵）★ウオッチドックタイマ:1ch（インターバルタイマとしても使用可）★DMAコントローラ:ショートアドレス4ch（最大）、フルアドレス2ch（最大）★リフレッシュコントローラ:2CAS/2WE方式（インターバルタイマとしても使用可）★A/D:10bit×8ch、D/A:8bit×2ch ★ I/O:入出力78本（最大）★16MBまでリニアアドレス拡張可 ★基本クロックをソフトで1/1～1/8まで分 ad 能）★周辺モジュール（タイマ、A/D、シリアル他）を個々にスタンバイ可（モジュー 能）★1MHz～16MHz動作可（1:1発振、発振回路内蔵、外部発振可）★5V、3 、3V動作時は8MHz動作まで）★5V常温域（20℃～30℃）で最大20MHz動作
AKI-H8開発キット（即使えるキット） ¥7,800 20P（10×2）ソケット×1、10P（5×2）ソケット×1、16×2 LCDモジュール付 （開発ソフト、アッセンブラーソフト、ライター一式付）、技術（ハード）資料集付、ソフト資料集付、ボード操作、アプリケーション集付、専用マザーボード付、DC19V、30Wスイッチングボード（電源）付
■AKI-H8/3048Fボード限定Cコンパイラ（説明書・技術資料一式付）〈WINDOWS95/98対応CDR版〉…… ¥2,000
■AKI-H8/3048Fボード用モニターデバッガ（説明書一式付）〈WINDOWS95/98対応CDR版〉………… ¥2,000
■3048Fマイコンボード用BASICコンパイラ WINDOWS 98/95版CDR ¥2,000 ■H8/（3048F）LSIのみ 1コ¥2,400
データ（CD-ROM）付 ★40P（20×2）×2、20P（10×2）×1、10P（5×2）×1、ピンソケット付CPUと232IC実装済み。40P（20×2）ソケット×2コ、20P（10×2）ソケット×1コ、10P（5×2）ソケット×1コ、16MHz水晶×1コ、40×2（80ピンヘッダ）×2本、リセット用IC抵抗、セラコンパーツ一式付 1キット¥3,800

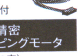

PIC BASICモジュール 20MHz動作 A/Dコンバータ内蔵
★PIC16F877に整数型BASICインタプリタ本体が書き込まれています。★付属の専用ソフトでエディット、デバック、書き込みまでできます。 ★面倒な液晶表示A/Dコンバータ、232C通信なども専用のコマンドで簡単に使えます。★ユーザープログラムはPICライターなしで何回も書き替えができます。
PIC BASICモジュール基板、専用ソフト（WIN98、ME,XP対応）CD付 1セット¥2,700

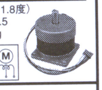

PICマイコン制御 ステッピングモータドライブキット
★正転・逆転OK ★8ビットマイコン制御で使い易く、精密安定にコントロール可 ★技術資料、専用ボード、C.R、パワーTR一式付
※モータ無しキット 1キット¥1,200


2相48ステップ小型ステッピングモータ
★48ステップ/相（7.5度）★軸3mm★外形4.2×2.3cm★重量110g
¥400技術資料付

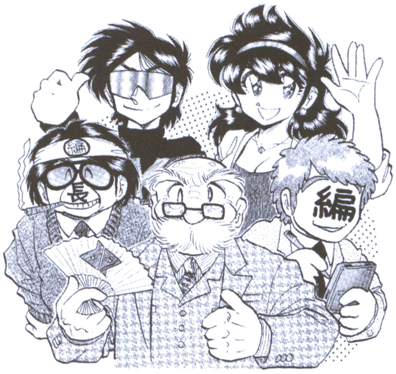




PICマイコン制御 ステッピングモータドライブキット
★7.5度（48ステップ）¥400ステッピング付キット★データ付
1キット¥1,500
PICマイコン制御 ハイパワーステッピングキット
★1.8度（200ステップ）¥700モータ付
★データ付 1キット¥1,700
2相200ステップ精密ハイパワーステッピングモータ
★200ステップ/相（1.8度）★軸8mm★外形5.5×5cm★重量350g
¥700 技術資料付

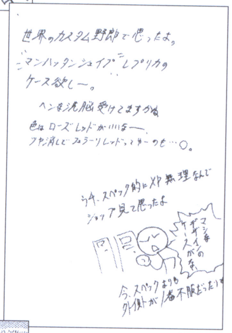




PICフラッシュマイコンIC
PIC16F84A-20/P …… 1コ¥380、10コ¥3,500
PIC16F873-20/SP（28ピンスリムDIP） 1コ¥800
PIC16F876-20/SP（28ピンスリムDIP） 1コ¥900
PIC16F877-20/P（40ピンDIP） …… 1コ¥1,000
PIC16F877-20PT（44ピンフラットIC） 1コ¥800
PIC16C620-20/SP（18ピンDIP） … 1コ¥290、10コ¥2,600
PIC16C621-20/SP（18ピンDIP） … 1コ¥300、10コ¥2,700
PIC16C711-20/P ……………… 1コ¥300、10コ¥2,700
PIC16C57C-20/P（28ピンDIP IC） 1コ¥240、10コ¥2,200
(株)秋月電子通商 http://www.akizuki.ne.jp
秋葉原本店 〒101-0021 東京都千代田区外神田1-8-3 野水ビル 1F/2F
TEL.03-3251-1779 FAX.03-3251-3357
代引通販で、さらにラクラクお買い物!! インターネット専用・通販センター
〒334-0063 埼玉県川口市東本郷252 TEL.048-287-6611 FAX.048-287-6612

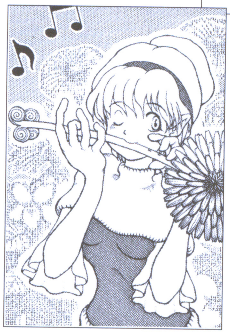
# あくせすROOM



◀福井県武生市・つかち～
編：最近はめっきり投稿が減ってしまいました。もっと気軽に，友達を誘って投稿をしてみよう！



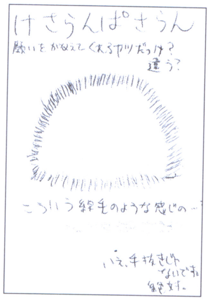
▶福島市・南斗オレガ
影：PCケースの中に，各種ゲーム機を入れてしまうのはどうだね。CDドライブは共用できないけど，場所をとらない。



◀愛知県名古屋市・わいち301
PS2のメモリカードが足りない？　スマートメディアで代用できる機械もあるみたいだよ。

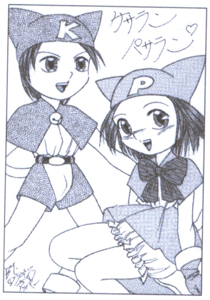
## うろ覚え選手権　　今月の課題「けさらんぱさらん」

つぐ美：「けさらんぱさらん」って知ってる？　私は，桐の箱に入った「白くてふわふわして，オシロイを食べて増える不思議な生き物」を見せてもらったことがあるの。でも，正直その正体は不明なのよね～。




◀福岡県山門郡・瞬間移動
影：これが一番，言い伝えに近いのかも。
つぐ美：夏でもふかふかしているのかな？

◀東京都練馬区・ルストフレム
つぐ美：ど，どんな味がするのかしら。
影：得体の知れないガスが吹き出してる……。




◀愛知県名古屋市・のうぎょうアッパー
つぐ美：ふわふわ飛んでいってしまいそうなコンビ名です。
編：左がケサランで，右がパサランですね。

## あくせすROOMの参加方法　〈投稿歓迎〉

あくせすROOMでは，みなさんの投稿をお待ちしています。愛読者カードや，官製ハガキ，または同サイズの丈夫な紙などを封書でまとめるなどして，お送りください。

鉛筆や色鉛筆はきれいに印刷されません。せっかくペンで描いてあっても，鉛筆の下書きが残っているものは，やはり載りにくくなりますので，インクや水性ボールペンなど，はっきりと描くける道具を使いましょう。文字だけの投稿であれば，ワープロを使ってもかまいませんよ（そのまま掲載できます）。

応募先
〒141-8715　東京都品川区東五反田1-11-15　電波新聞社
マイコンBASICマガジン編集部　「あくせすROOM」係

募集物の一例

●うろ覚え選手権
6月号の募集：道の終わりにはなにがあるのか？　海を行くところまで行き着くとなにがあるのか？　誰もたどりついたことのない「世界の果て」を想像力を駆使して描いてみましょう。4月15日必着です。
7月号の募集：あなたが振り向く直前まで，背後に存在していた世界を想像してみてください。鏡越しだと見えない「なにか」が，そこにはあるはずです。5月15日必着です。

●世界のカスタム野郎
愛用の道具をいろいろカスタマイズ（自己改造）して使っている人の自慢話を募集します。冷蔵庫にパーツを組み込んでDOS／Vパソコンを作る！　ゲームコントローラに新しく機能をつける！　ちょっとした工夫で，実用性がアップしたり，カッコ良くなったという実例を，ハガキ1枚をメドに紹介してください。

●デジカメ激写ボーイズ
デジカメを持って街に出よう！　デジカメやデジビで撮影したユニークなお笑い画像を送ってください。タイトルはアレですが，女性のかたの投稿も歓迎です。

あくせすROOMは，みなさんのパワーで作るページです。このほかにも，おもしろいネタがあったら，文字でもイラストでも，思いつくままに表現してみてください。企画として長くつづくポイントは，ほかの人がちょっとしたアレンジで相乗りできること。みんなで楽しく参加できそうなネタを常時募集しています。

## ソニーが小型二足歩行ロボットを開発

**高度な運動，豊かなコミュニケーションを実現！**

ソニーは3月19日，家庭の中で想定される環境や場面に適応して行動する小型二足歩行エンターテインメントロボット『SDR-4X』を開発したと発表しました。

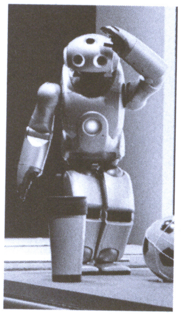
ソニーは，2000年11月にパラパラを踊る二足歩行ロボットを開発しましたが，今回のSDR-4Xは，各種センサ・システムと記憶や学習に基づく行動制御により，高度な運動性能を持たせると同時に，人間とのコミュニケーションができるようにしたのが特徴です。

各関節を駆動する小型アクチュエータの出力性能を向上させるとともに，内蔵された各種センサからの情報に基づいて全身38か所の関節部をリアルタイムで制御する「実時間統合適応制御システム」を新たに開発しました。これにより，不整地面や傾斜面での二足歩行だけでなく，外部から力が加えられた場合でも姿勢を保持するとともに，歩幅や旋回角度など，状況に応じて必要な歩行動作パターンをリアルタイムで生成することで，安定歩行を可能にしています。

また，搭載されている二つのCCDカメラの視差を利用し，被写体までの距離を検出，床面の存在や障害物とロボットとの間隔などを認識し，障害物を避けた経路を自動的に生成して歩行することもできます。

さらに，画像・音声認識技術や音声生成技術に加えて，記憶に基づく対話や行動の制御技術を取り入れ，CCDカメラに入力された正面顔画像から顔を検出，誰であるかを識別します。頭部に配置された七つのマイクロフォンを使用して，音のする方向を検出するとともに，話者識別を行なうことも可能です。

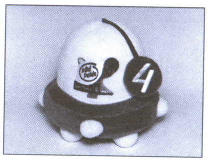
今回の製品は商品化に向けたプロトタイプですが，年内には商品化の方向性が見えるかもしれないということですので，楽しみです。価格のほうは高級自動車1台分とか。



試作機"SDR-4X"の主な仕様
CPU：64ビットRISCプロセッサ（×2）
主記憶装置：64MB DRAM（×2）
OS：Aperios（ソニー独自のリアルタイムOS）
ロボット制御アーキテクチャ：OPEN―R
制御プログラム供給媒体：16MBメモリースティック

## 5月号お楽しみプレゼント

①モバイル インテル Pentium4プロセンサ-M発売記念　インテル特製の癒しグッズ，竹炭入り「インテル モバイルUFO抱き枕」／5名（提供:インテル）



②書籍「オールアバウト・カプコン対戦格闘ゲーム1987-2000」特製クリアファイル付き／5名（提供：電波新聞社）

●応募方法：本誌とじ込み愛読者ハガキ（またはインターネット読者ハガキ）にプレゼント記入欄があります。一つだけ選んでお送りください。
●締め切り：2002年5月7日必着
《2002年3月号の当選者》①北海道天塩町・宮西和機さん　②愛知県名古屋市・伊藤隆文さん他4名　③-1大阪市・広岡聡之さん他4名　③-2大阪府茨木市・吉門純一さん他4名　③-3青森県弘前市・奥野健太郎さんさん他4名

# Falcom® NEWS EXPRESS

日本ファルコム株式会社
〒190-0012 東京都立川市曙町1-14-13
TEL 042-527-6501
FAX 042-528-2714

スタッフ募集中
採用担当BFM係
http://www.falcom.co.jp/

# Zwei!!に続くファルコム最新作の主要キャラクターを一挙紹介

速報

## 主要キャラクター挙紹介！

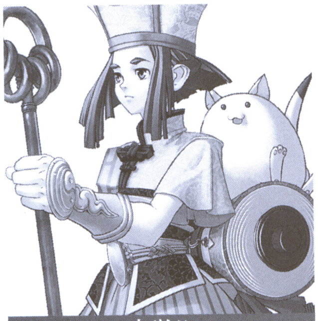
ヒミカ

モモスケ

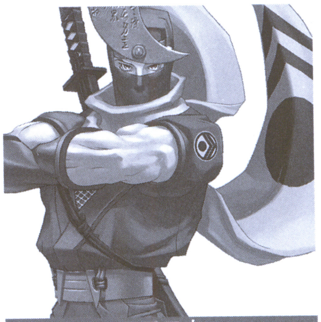
キクカ

ロレンゾ

シュレン


セツハ

タクワン

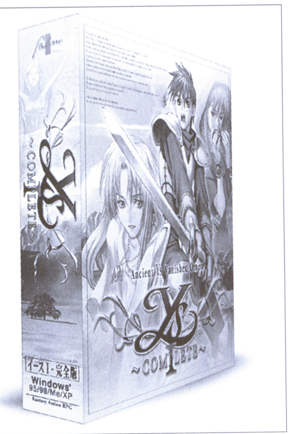

ロフロウ

ファルコムファンの皆様，お待たせしましたっ！ついにZwei!!に続く最新作が登場します。タイトルはまだ秘密ですが，新作に登場する主要キャラクターを一挙に紹介します！すでにファルコムホームページではいくつか公開されているけど，次の新作は，名前や姿を見てわかるように「和」風となっています。しかも，キャラクターそれぞれに独自のシナリオが用意されていますので，誰を選ぶかによって展開が変わるかも？とりあえず今回は8人だけだけど，実は隠しキャラがいるという噂も……。これからどんどん新情報が公開されていくのでお楽しみに！

最新情報はファルコムホームページで！
http://www.falcom.co.jp/

## XP対応「イースI完全版」好評発売中

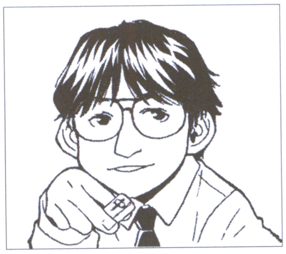
ファルコム通販価格：3800円



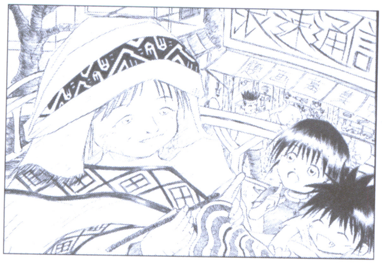
パソコンRPGの定番中の定番「イース」が，新価格・新パッケージで好評発売中！しかも初回版は、3つのファルコムNEWグッズがオマケに付いてるので購入される方はお早めに！
(オマケのグッズは別売りもあります)

3つのオマケ　※詳細はファルコムHPで！

① ファルキャラトレーディングコースター(別売480円)
② イースたためるＣＤスタンド　(別売1200円)
③ イースチビデカ２way マット (別売480円)



新作のテスト版をプレイしていますが，かなり面白くハマっています。選んだキャラはロクロウですが，なかなか手応えがあっていいです。早く情報を知りたいっ！と思っている皆さん，ただ今準備中ですのでもうしばらくお待ちください。突然HPで発表されるかもしれませんので，常にチェックしておきましょう！



## 支持率低下？
# なにわ通信

PRESENTED BY マイコンソフト ADDRESS● 〒530-0005大阪市北区中之島3-2-4 朝日新聞ビル6F電波新聞社大阪本社内 マイコンソフト株式会社 なにわ通信係

http://www.micomsoft.co.jp

タイトル冠文：山形県・御堂筋子
タイトル・イラスト：大阪府・暁様へ
本文：暁

皆さんこんにちは。やっぱり春は良いですねえ～。私は，またもや骨にヒビが入ってしまいました。

ちょっと重いディスプレイを担ぎ上げただけで，背中のアバラ骨を痛めてしまいました。

普段運動していないからか，食生活が乱れていたからはよく分かりませんが，最近，よく骨を痛めます。

1年半で3回もアバラ骨にヒビが入る人なんて，そんなに居ないでしょうね。

## 南国のお話

最近，仕事で台湾に行くことが多くなってきました。

今回は台湾特集をしてみたいと思います。

マイコンソフトの商品は全部台湾で生産されていますので，開発や生産の打ち合わせで，どうしても台湾(台北や高雄)に行く機会が増えてきます。

台北は日本から飛行機で2時間半程度。

台湾は南国なので，冬でも当然暖かいです。

夏は蒸し暑く，死にそうです

台北の街で，まず驚くのがバイクの多さです。特に通勤時間帯は，スクーターが道路を占領します。歩行者は横断歩道もスクーターを避けて渡らなければならず，とても危険です。(大阪の100倍危険です)

また，タクシーもとても激しい運転をするので，ジェットコースターに載っている気分です。とりあえずクラクションを鳴らすのは挨拶がわりといったところでしょうか。

台湾は日本以上に車中心の社会です。

## 台湾は漢字だから

ご存じの通り台湾でも漢字を使います。

これは，日本人にとって非常に助かります。

例えば「学校」は「學校」と書きますので，見れば何の建物かすぐに分かります。

ホントに漢字は便利ですよ。

もちろん難しい漢字も多く，そういった漢字は読めませんけどね(笑)。

## 台湾の大衆文化

日本人が洋楽を聴くように，台湾では日本のポップスや洋楽をよく聴きます。例えば「浜崎あゆみ」とか「宇多田ひかる」などは台湾でも有名です。

もちろん台湾の歌手も日本人と同じようにステキな方が多く，歌も旨いので念のため。

また，台湾ではCATVが主流で，チャンネルも90局以上あります。日本のNHK BSなども受信可能になっていることもあります。

日本のアニメも台詞が吹き替えで放映されていたり，日本のドラマやバラエティー番組が日本語のまま放映されています。(画面の下部に必ず漢字の字幕が表示されます。)

## 日式

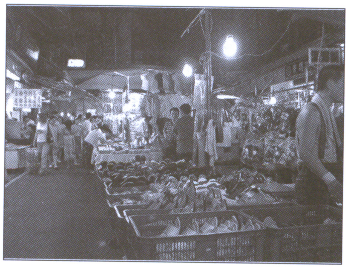
台湾では「日式・・・」というような看板を良くみかけます。これは「日本式」という意味です。台湾では和食も浸透しているようです。

ラーメンなども「日式」の物が流行っています(はっきり言って美味しい)。

## 次号も

次号は台湾特集，第2弾をお送りします。お楽しみに。

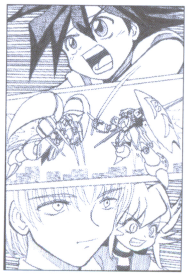
[写真1]泊まったホテルの近くで夜市が行われていると聞いて，行ってみました。



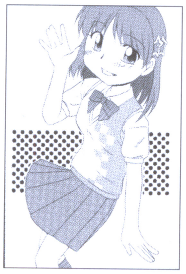
[写真2]夜市の様子です。たいへん賑やかで，色々な物を売っています。

[写真3]携帯電話の本体ケースを売っています。台湾では，ケースを交換するのが流行り。

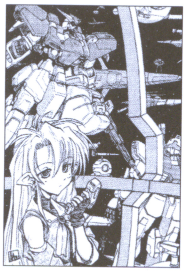
### ありがたい，おハガキのコーナー



埼玉県・野沢プロト
　投稿者の栽刃鋭人さんがデザインされた「レアファイター月人」というメカだそうです。カッコイイメカですね！



埼玉県・ぶんちょ
　春ですねー！春は入学シーズンですねー！初々しい制服を着た人たちを見ると思わず微笑んでしまいますよね。



東京都十大阪府・KARASU+M
　KARASUさんとマヲさんの合作です。女の子担当がKARASUさんで，メカ担当がマヲさんです。・・・流石ですね。

# IT新発見!! ITが僕につながる世界につながる

## ネットプログラマーコース

ネットを通じて人々がつながるインターネット。ビジネス・教育・趣味・情報検索など、様々な利用方法があり、インターネットを利用する一般ユーザーやビジネスに活用する企業は増加する一方です。それによって当然ホームページの充実度が求められ、人材の需要も急増しているというのが、この業界の現状です。ネットプログラマーコースでは、ホームページを制作するためのコンピューターの基礎知識をはじめ、ビジュアルBASIC・ビジュアルC++・JAVAプログラミング等のプログラマーに欠かせない技術と知識の修得を中心に、Webデザインの制作必要となる全ての要素を基礎から学び、本来のデザイナーの仕事の枠を越えたエンターテイメント性のあるWebアートディレクターとしての活躍も可能となる技術を修得するコースです。

## CG&Web デザイナーコース

ホームページ制作技術力が各業界で求められている事は間違いありませんが、これからのWebデザイナーに求められているのは単なる技術力のみではなく、いかに充実度を高められるかというデザインのセンスです。CGはデジタル業界に限らず、様々な業界で必要とされている技術で、ITの急激な成長に伴い、CG制作の技術を持つ人材が多く求められています。このコースでは、ホームページ及びCG制作に必要な要素を基礎から学ぶコースです。

**2003年4月生募集中!!**

入学案内は無料です。お気軽にお電話ください。
体験入学・学校見学会は随時実施中です。

# IT ネットクリエイター学院 東京校・大阪校・名古屋校

【IT ネットクリエイター学院 東京校】〒151-0053 渋谷区代々木3-57-6 ☎03-3378-8150【IT ネットクリエイター学院 大阪校】〒530-0041 大阪市北区天神橋1-12-6 ☎06-4801-4710
【IT ネットクリエイター学院 名古屋校】〒453-0801 名古屋市中村区太閤4-9-26 ☎052-452-6881

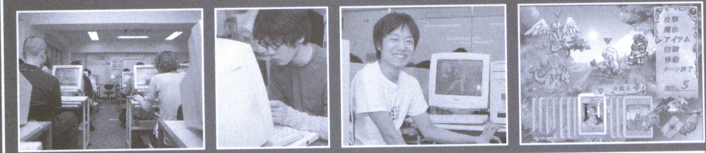
### ゲームデザイナー養成講座

●全日制 月曜日～金曜日 AM10:00～PM4:00

**1年コース**
1年間でデザインからプログラムの制作まで、ゲーム制作の一連の流れを全てマスターするコースです

**2年コース**
デザインからプログラムまで2年間じっくり勉強できます。時間がある為、凝ったゲームを制作できます。

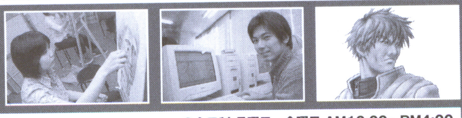
**3年コース**
このコースは3年間かけて、かなり高度で未来的な技術も併せて修得する事を志す方には最適です。



### ゲームキャラクターデザイナー養成講座

●全日制 月曜日～金曜日 AM10:00～PM4:00

**1.2年コース**
デッサンの基礎からキャラクターのデザイン・キャラクターのアニメーション技術を実習中心の授業体制で徹底指導します。



### CGクリエイター養成講座

●全日制 月曜日～金曜日 AM10:00～PM4:00

**1.2.3 年コース**
デッサンの基礎から2D・3Dグラフィック及び2D・3Dグラフィックアニメーション技術を実習中心に完璧指導します。

### 3Dグラフィック&モーションキャプチャークリエイター養成講座

●全日制 月曜日～金曜日 AM10:00～PM4:00

**2年コース**
デッサンの基礎からLight-wave、Softimageを用いて3Dグラフィック技術、最新技術「モーションキャプチャー」まで勉強します。

### 企画・シナリオライター養成講座

●全日制 月曜日～金曜日 AM10:00～PM4:00

**1.2.3 年コース**
ゲーム企画、シナリオを中心にゲーム心理学、CG、キャラクターデザイン、プログラミングを勉強する本格的な養成講座です。

### サウンドクリエイター養成講座

●全日制 月曜日～金曜日 AM10:00～PM4:00

**1.2年コース**
サウンドクリエイターを目指す方の為に設けられた、ゲームの作曲からサウンドドライバー作成まで学習する一貫教育講座です。

### ゲーム&ネットプログラマー養成講座

●全日制 月曜日～金曜日 AM10:00～PM4:00

**2年コース**
ゲーム制作に加え、インターネット制作技術を学ぶ講座です。デジタル業界全般で役立つ技術を身に付け就職を目指します。

## ゲーム会社への就職者数5年連続100名突破!

当学院ゲームクリエイター科卒業第7期生の2002年卒業生がご覧のようにゲーム会社にプログラマー等の専門職・正社員として大量就職を果たしました。業界の常識を大きく上回るゲームスクール始まって以来の5年連続100名突破という驚異的な実績です。独自のカリキュラム・指導方針・そして最先端の環境が生んだ結果です。

【2002年3月卒業生の主な就職先】ナムコ、カプコン、テクモ、フロムソフトウェア、ユークス、CSK総合研究所 SEGA-AM2、セガ・ロッソ、スクウェア、アイディアファクトリー、サクセス、元気、データウエスト、サンライズ、OLMデジタル、クレイジーゲーム、ナツメ、ゲームフリーク、タムソフト、アルトロン、タクミコーポレーション、インテリジェントシステムズ、ラクジン、アルファシステム、テクニカ、拓洋興業、プロハウス、Qゲーム、フライトプラン、ジーアーティスツ、A.I、フォレストエンターテイメント、童、AXE、ハイウェイスター、クールキッズ、MIT その他多数（一部アルバイト入社含む）

# 東京・大阪・名古屋ゲームデザイナー学院

【東京ゲームデザイナー学院】〒151-0053 渋谷区代々木3-57-6 ゲームクリエイター科 ☎03-3370-2720 アニメ・マンガコミック科 ☎03-5350-2422
【大阪ゲームデザイナー学院】〒530-0041 大阪市北区天神橋1-12-6 ☎06-6882-4980 【名古屋ゲームデザイナー学院】〒453-0801 名古屋市中村区太閤4-9-26 ☎052-452-5901

# 中古通販

PC NET ピーシーネット

在庫保有数日本最大級
(常時10,000台)

# TEL.03-3733-1441

通販営業時間／月～金 9:00～19:00 土・日・祝日 11:00～19:30

●アドレスが新しくなりました

## www.pc-net.jp/

今までのアドレスでも引き続きご利用いただけます。

**www.prins.co.jp/**

楽天市場店 http://www.rakuten.co.jp/pcnet/
ジャストネット http://emall.justnet.ne.jp/shop/pcnet/
ネットプライスモール http://www.netpricemall.com/
ビッグローブショッピングワールド http://shopping.biglobe.ne.jp/

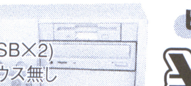
## 最新情報は電話かネットでチェック!!

急げ!!

## 今月の目玉!!

御注文の際は、必ず「雑誌を見た!!」を合い言葉で御願い致します。

NEC U10151
PC-MA30D/M-43
(Celeron(300)/64MB/4.3GHD/CD32/USB×2)
外装ヤケ/外装塗装剥げ/キーボード無・マウス無し
※キーボード・マウスはお客様のご用意となります。



¥18,800↓
ビックリ価格
**¥8,600**

1週間保証
在庫なくなり次第終了

インターネットでは毎回お得な企画がいっぱい!!
3日間保証セール だから安い!!
期間限定タイムセール

## 全商品 外装クリーニング 無料!サービス

[安心の全品保証付] 1万円以上…1ヵ月保証 1万円以下…1週間保証

※本商品は添付品(OSを含む)は付属しておりません。
※中古パソコン販売時の在庫定義が変わりました。
今後在庫とは、従来品に加え、動作上支障のないキズ・天板やけ・カバー等の欠品がある物や、キーボード等が補充(他メーカー含)されている商品も、「在庫」としてお取り扱いさせていただきます。

### DOS/V

| 注文No. | 商品 | 価格 |
|---|---|---|
| U8344 | Aptiva(2190-27N)(IBM)(K6-2(500)/60MB/10GHD/CD/擦り傷/外装焼け/キーボード・マウス無/省スペース型/PCIX3/USBX2) | ¥29,800 |
| U1486 | Aptiva(2196-47J)(IBM)(K6-2(533)/56MB/15GHD/CD40/外装焼け/擦り傷/省スペース型/PCIX3/USBX4) | ¥31,000 |
| U8345 | Aptiva(2196-47N)(IBM)(K6-2(533)/56MB/15GHD/CD/外装ヤケ/擦り傷/外装塗装剥げ/省スペース型/PCIX3/USBX4) | ¥31,000 |
| U8777 | Aptiva(2197-6JJ,6HJ)(IBM)(Celeron(633)/64MB/40GHD無し/CD48/外装ヤケ/HDD無し/省スペース型/PCIX3/USBX4) | ¥24,800 |
| U10077 | PC300 GL(6561-55.56J)(IBM)(PenII(300)/96MB/4.2GHD/CD32/擦り傷/MO付き(動作未チェックの為、保証無し)PCI×2/ISA×1/共有×1/USB×2) | ¥24,800 |
| U10156 | PC300 XL(6588-73J)(IBM)(PenII(300)/64MB/4.2GHD/CD16/外装焼け/擦り傷/ISAX2/共有X3/USBX2) | ¥23,800 |
| U8938 | PC300 PL(6862-B1J/3J/4J)(IBM)(Celeron(300A)/64MB/6.4GHD/CD32/外装焼け/ゴム足無/ファイルカバー無し/PCIX2/ISAX1/共有X1/USBX2) | ¥16,800 |
| U8744 | NETVISTA A40(6881-20J/22J/S7J)(IBM)(Celeron(600)/64MB/10GHD/CD24/外装焼け/擦り傷/スタンド無し/省スペース型/USBX2) | ¥49,800 |
| U1981 | PC-MA26DS…41(NEC)(PenII(266)/128MB/4.3GHD/CD24/LAN/外装焼け/汚れ/擦り傷/マウス無し/32MB増設/PCIX4/USBX2) | ¥24,800 |
| U10161 | PC-MA33H/S…43(NEC)(Celeron(333)/64MB/4.3GHD/CD32/擦り傷/外装焼け/汚れ/マウス無/PCIX4/USBX2) | ¥27,800 |
| U1001 | PC-MA23CS(NEC)(MP(233)/128MB/3.2GHD/CD24/LAN/外装ヤケ/汚れ/マウス無し/PCIX4/USBX2) | ¥12,000 |
| U10021 | PC-MA30D/M-41(NEC)(PenII(300)/96MB/4.3GHD/CD24/外装焼け/擦り傷/ミドルタワー型/PCIX5/USBX2) | ¥23,800 |
| U10185 | PC-FN23CCXAAL1(液晶一体型)(NEC)(MP(233)/96MB/2.1GHD/CD16/14.1TFT(1024X768)/一体型/外装焼け/擦り傷/USBX2) | ¥29,800 |
| U2116 | DESKPRO EP6300A(COMPAQ)(Celeron(300)/64MB/6.4GHD/CD32/外装焼け/ゴム足(1ヶ)/タワー型/デスクトップ型共用/PCI×3/ISA×2/USB×2/AGP×1) | ¥21,800 |
| U10174 | DESKPRO EXS P733(COMPAQ)(PenIII(733)/128MB/20GHD/CD24/省スペース型/PCIX2/USBX2) | ¥54,800 |
| U10158 | OPTIPLEX GXA(DELL)(PenII(233)/128MB/3.2GHD/CD/外装焼け/擦り傷/汚れ/高さ:10.9cm) | ¥13,800 |
| U1135 | OPTIPLEX GXA(DELL)(PenII(266)/64MB/3.2GHD/CD/外装焼け/マウス無し/高さ:10.9cm) | ¥13,800 |
| U10127 | OPTIPLEX GXA(DELL)(PenII(300)/64MB/6.4GHD/CD24/外装焼け/汚れ)… | ¥19,800 |
| U1729 | OPTIPLEX G1(DELL)(Celeron(400)/64MB/4.3GHD/CD32/PCIX1/ISAX1/USBX2) | ¥26,800 |
| U10062 | OPTIPLEX GX1(DELL)(PenII(450)/128MB/8.4GHD/CD/マウス無し/外装キズ/高さ:16.5cm/PCIX2/ISAX2/共有X1/USBX2) | ¥29,800 |
| U10027 | OPTIPLEX GX100(DELL)(Celeron(500)/256MB/6.4GHD/CD/省スペース型) | ¥39,800 |
| U10180 | Dimension XPS D233(DELL)(PenII(233)/128MB/4.3GHD/CD/擦り傷/外装焼け/ミドルタワー型) | ¥12,800 |
| U8907 | Dimension XPS D300(DELL)(PenII(300)/64MB/3.2GHD/CD/外装焼け/ミドルタワー型) | ¥17,800 |
| U10164 | Dimension XPS D333(DELL)(PenII(333)/126MB/4.3GHD/CD/擦り傷/外装焼け/ミドルタワー型) | ¥19,800 |
| U8840 | GATE WAY 2000(G6-300M)(GATEWAY)(PenII(300)/64MB/6.4GHD/CD-R(未チェックの為保証無し)/外装焼け/スロットカバー無し/ミドルタワー型:47cm) | ¥19,800 |
| U10066 | GATE WAY 2000(GP6-333J)(GATEWAY)(PenII(333)/64MB/6.4GHD/CD/PCIx3/共有x1/AGPx1) | ¥21,800 |
| U1202 | GATE WAY 2000(GP6-333)(GATEWAY)(PenII(333)/128MB/8.4GHD/CD32/外装焼け/ミドルタワー型(47cm)/PCI×3/ISA×1/共有×1/USB×2/AGP×1) | ¥23,800 |
| U8354 | GATE WAY 2000(GP6-350)(GATEWAY)(PenII(350)/64MB/6.4GHD/CD/高さ:48cm) | ¥23,800 |
| U10101 | Vectra VL6/350-8(HP)(PenII(350)/64MB/4.3GHD/CD/外装焼け/擦り傷/汚れ) | ¥22,800 |
| U10169 | Vectra Vei7/466(HP)(Celeron(466)/128MB/4.3GHD/CD32/擦り傷/外装焼け/外装キズ/PCIX3/ISAX1/USBX2) | ¥33,800 |
| U1968 | FMV-6233D8(FMV-623D8C5)(富士通)(PenII(233)/64MB/3.2GHD/CD16/外装焼け/汚れ/32MB増設/PCIX2/ISAX2/USBX3) | ¥11,800 |
| U8579 | FMV-1DX7101,7151(FMV-6266DX)(富士通)(PenII(266)/64MB/3.2GHD/CD24/擦り傷/外装焼け/PCIX2/ISAX1/USBX2) | ¥16,800 |
| U8642 | FMV626T6C5,7共通(富士通)(PenII(266)/64MB/4.3GHD/CD/擦り傷/外装焼け/タワー型/PCIX3/ISAX1/共通X1/USBX2) | ¥14,800 |
| U1071 | FMV7T90B11.61(FMV6300T7)(富士通)(PenII(300)/32MB/4.3GHD/CD20/外装焼け/擦り傷/タワー型/PCIX3/ISAX1/共有X1/USBX2/AGPX1) | ¥21,000 |
| U10183 | FMV-7TB0B11.61(FMV-6333T7)(富士通)(PenII(333)/192MB/4.3GHD/CD20/汚れ/擦り傷/外装焼け/ミドルタワー型/PCI×3/ISA×1/共有×1/USB×2/AGP×1) | ¥25,800 |
| U20221 | PC1DC6-E4222HCOO(液晶一体型)(日立)(PenII(333)/64MB/4.3GHD/CD24/14.1TFT(1024X768)/外装焼け/省スペース型/PCIX1/USBX2) | ¥39,800 |
| U10197 | HA80000/30(GTD030A3-H3C1N10)◆省スペースサーバー◆使用期間6ヶ月の美品(日立)(PIII(800)/128MB/40GHD(アレイ)/CD24/スタンド無し/PCI×3/USB×2) | ¥99,900 |
| U10191 | I DATEN(オーテック)(Celeron(400)/64MB/10GHD/CD40/外装焼け/擦り傷/ミドルタワー型/PCIX3/USBX2) | ¥22,800 |
| U10189 | I CLASS 400V(オーテック)(Celeron(400)/64MB/10GHD/CD/外装ヤケ/擦り傷/ミドルタワー型/PCIX3/USBX2) | ¥22,800 |

### DOS/Vノート

| 注文No. | 商品 | 価格 |
|---|---|---|
| U20147 | ThinkPad 240(2609-31J,35J)◆Win98SE付き(IBM)(Celeron(366)/192MB/6.4GHD/FDコネクターカバー無/PCダミーカバー無/擦り傷/純正箱付き/10.4TFT/USB×1/B5サイズ) | ¥76,800 |
| U20057 | ThinkPad 315ED(2600-DZJ,DOJ)(IBM)(MP(166)/32MB/2.1GHD/CD20/SVGA/擦り傷/AC型番違い/12.1TFT) | ¥39,800 |
| U3494 | ThinkPad 360CS(2620-2JF)(IBM)(486SX(33)/4MB/340HD/VGA/PCダミーカバー無し/AC型番違い/9.5DSTN) | ¥4,000 |
| U6554 | ThinkPad 380D(2635-3AJ)(IBM)(MP(150)/48MB/2.1GHD/CD8/SVGA/外装塗装剥げ/ゴム足無し/12.1TFT) | ¥33,800 |
| U6504 | ThinkPad 380ED(2635-6A,YJ)共通(IBM)(MP(166)/32MB/3GHD/CD20/SVGA/ドット抜け/リアカバー無し/12TFT) | ¥32,800 |
| U6303 | ThinkPad 380XD(2635-9AJ)(IBM)(MP(233)/64MB/4GHD/CD24/SVGA/擦り傷/外装焼け/AC型番違い/12TFT) | ¥49,800 |
| U3864 | ThinkPad 560(2640-EJA/EJ9)(IBM)(Pen(120)/8MB/810HD/SVGA/擦り傷/外装焼け/12.1TFT) | ¥12,800 |
| U20229 | ThinkPad 560X(2640-70J)(IBM)(MP(233)/48MB/4GHD/SVGA/外装ヤケ/擦り傷/AC型番違い/12.1TFT) | ¥32,800 |
| U20155 | ThinkPad 570(2644-3AE)(IBM)(PenII(366)/64MB/6GHD/擦り傷/リアカバー無し/13.3TFT) | ¥69,800 |
| U3160 | ThinkPad 755C(9545-FJ8)(IBM)(486DX2(50)/12MB/330HD/ACアダプタ無し/外装焼け/外装カケ/PCカバー無し/10.3DSTN) | ¥4,600 |
| U20108 | PC-VP16CWXAA1(NEC)(MP(166)/48MB/2.1GHD/CD20/XGA/擦り傷/CDデバイスドライブ無し/13.3TFT/USBX1) | ¥32,800 |
| U6265 | PC-LV16C/WSDAF1(NEC)(MP(166)/32MB/2.1GHD/CD20/SVGA/モデム/擦り傷/外装焼け/デバイスドライバ無し/12.1TFT/USBX1) | ¥37,800 |
| U20029 | ARMADA 1520DM(COMPAQ)(Pen(133)/32MB/1GHD/CD10/SVGA/擦り傷/11.3DSTN) | ¥16,800 |
| U6928 | ARMADA 4160T(COMPAQ)(MP(166)/48MB/2.1GHD/SVGA/擦り傷/キーボード文字焼け/CDドライブ付き/11.8TFT/CDオプション) | ¥39,800 |
| U3585 | LATITUDE XPI P120ST(DELL)(Pen(120)/8MB/810HD/10.4TFT)……… | ¥9,800 |
| U3927 | FMV-52NA2J1,5共通(富士通)(Pen(120)/32MB/810HD/CD/SVGA/外装焼け/擦り傷/11.3TFT) | ¥13,800 |
| U6632 | FMV-NC13D(富士通)(Pen(133)/32MB/1GHD/モデム/VGA/外付けFDD/外装焼け/外装塗装剥げ/7.2DSTN) | ¥15,800 |
| U20008 | FMV-53NU6Y1,6(富士通)(Pen(133)/32MB/1.3GHD/SVGA/擦り傷/外装焼け/12.1DSTN) | ¥10,000 |
| U3941 | FMV-53NU5S(富士通)(Pen(133)/32MB/1GHD/CD10/モデム/SVGA/擦り傷/11.2TFT) | ¥26,800 |
| U3483 | FMV-53NP6W5(富士通)(MP(133)/32MB/1.6GHD/CD16/SVGA/バッテリーパック無し/12.1TFT) | ¥27,800 |
| U3925 | FMV-NU151,152,153共通(富士通)(MP(150)/32MB/1.6GHD/CD/SVGA/擦り傷/外装焼け/12.1TFT) | ¥32,800 |
| U20133 | FMV-56NA7W5,7(富士通)(MP(166)/64MB/2.1GHD/CD20/SVGA/擦り傷/外装ヤケ/キーボード文字剥げ(4)/12.1TFT) | ¥33,800 |
| U20149 | FMV-NU516D1,2,3(富士通)(MP(166)/32MB/2.1GHD/CD20/SVGA/擦り傷/外装塗装剥げ/12.1DSTN/USBx1) | ¥29,800 |
| U6343 | FMV-56NA7W5,7(FMV-5166NA/W)(富士通)(MP(166)/32MB/2.1GHD/CD20/SVGA/LAN/外装焼け/擦り傷/12.1TFT/USBX2) | ¥36,800 |
| U20194 | FMV-2NU6WB0,6,5 1(富士通)(PenII(233)/64MB/3.2GHD/SVGA/外装焼け/擦り傷/12.1TFT) | ¥34,800 |
| U20232 | FMV-2NA6LC0.5,65共通(FMV-6233NA2/L)(富士通)(PenII(233)/96MB/3.2GHD/CD24/擦り傷/外装焼け/14.1TFT/USBX2) | ¥49,800 |
| U20233 | FMV-3NA7LC0,6,5共通(富士通)(PenII(266)/96MB/4.3GHD/CD24/擦り傷・外装焼け・キーボード文字剥げ/14.1TFT/USBX1) | ¥54,800 |
| U20234 | DynaBook PA1257C9(東芝)(MP(233)/96MB/4GHD/CD20/SVGA/外装焼け/ドット抜け/12.1TFT/USBX1) | ¥37,800 |
| V0188 | PC1NH5-E4D22R320(FLORA270EX)◆Win98付(日立)(PenII(333)/64MB/4.3GHD/CD20/LAN/活用百科CD-ROM/ドットカケ(4)/13.3TFT/付属品付) | ¥67,800 |
| U3586 | CF-35EJC4(パナソニック)(Pen(133)/16MB/1.44GHD/SVGA/擦り傷/PCダミーカード無し/12.1TFT) | ¥15,800 |
| U20211 | M3455-A1CO(三菱)(MP(233)/48MB/3.2GHD/SVGA/13.3TFT)……… | ¥34,800 |

バッテリーパックは消耗品につき保証対象外とさせていただきます。

### 98デスクトップ

| 注文No. | 商品 | 価格 |
|---|---|---|
| U0210 | PC-9801FA/U2(486SX(16)/1.6MB/FM/マウス無し/外装焼け/擦り傷)…… | ¥27,800 |
| U2634 | PC-9821V233/M3(MP(233)/32MB/3.2GB/CD20/Fax36.6K/擦り傷/汚れ/外装焼け/スピーカー・マウス無しミニタワー型) | ¥13,800 |
| U2064 | PC-9821XB10/J8(Pen(100)/16MB/850HD/外装焼け/マニュアル有り)…… | ¥19,800 |
| U1636 | PC-9821XC16/S5A3,S5B3◆Win95付き(Pen(166)/32MB/1.6GHD/CD8/外装焼け/擦り傷/添付品付) | ¥14,800 |
| U8610 | PC-9821RA223/N30(PenII(233)/64MB/3GHD/CD16/LAN/外装焼け/外装キズ/PCIX2) | ¥27,800 |
| U8585 | PC-9821RA223/W30(PenII(233)/96MB/3GHD/CD16/外装焼け/外装キズ/PCIX2) | ¥29,800 |
| SH801 | PC-9821RA223/N30◆NT4.0(PenII(233)/64MB/3GHD/CD16/付属品付)‥ | ¥39,800 |
| U8352 | PC-9821RV226/N20(PenII(266)/96MB/2GHD/CD6/外装焼け/PD付き/ミドルタワー型) | ¥54,800 |
| U8793 | PC-9821RA266/W30R(PenII(266)/96MB/3GHD/CD16/外装焼け/擦り傷)‥ | ¥38,800 |
| U8996 | PC-9821RA300/M40.W40.D40(Celelon(300)/64MB/4GHD/CD24/LAN/外装焼け/擦り傷) | ¥42,800 |
| U10203 | PC-9821RA40W/60CZ(Celelon(400)/96MB/6.4GHD/CD24/外装焼け/PCIX2) | ¥59,800 |

### 98ノート

| 注文No. | 商品 | 価格 |
|---|---|---|
| U3047 | PC-9821ND/340(486SX(33)/5.6MB/340HD/キーボード文字焼け/外装焼け/擦り傷/9.5DSTN) | ¥12,000 |
| U3420 | PC-9821ND(8/340)(486SX(33)/8MB/340HD/キーボード焼け/擦り傷/9.5DSTN/メモリ・HD増設) | ¥11,800 |
| U3176 | PC-9821NE(486SX(33)/14.6MB/340HD/110pin拡張バス/キーボード文字焼け/外装焼け/擦り傷/9.5TFT) | ¥17,800 |
| U3273 | PC-9821NE3(486DX2(50)/1.6MB/外装焼け/擦り傷/ゴム足無し/10.1TFT)… | ¥13,300 |
| U0345 | PC-9821NE3/3(486DX2(50)/8MB/350HD/擦り傷/ドット抜け/10.1TFT)…… | ¥14,800 |
| U3334 | PC-9821NF/810(Pen(75)/7.6MB/810HD/PCM/メモリカバー無し/9.5TFT)… | ¥13,800 |
| U6237 | PC-9821LA10/5A,B共通(Pen(100)/24MB/540HD/VGA/サブノート/PCダミーカード無し/外装焼け/9.5TFT/B5サイズ) | ¥16,800 |
| U2766 | PC-9821NA12/H8(Pen(120)/16MB/810HD/CD/PCM/バッテリーパック欠品/擦り傷/10.4TFT) | ¥16,800 |
| U3557 | PC-9821NA15/X14(Pen(150)/16MB/1.4GHD/CD8/PCM/外装焼け/擦り傷/リアカバー無し/12.1TFT) | ¥31,800 |
| U6750 | PC-9821NR13/D10A(Pen(133)/32MB/1GHD/CD/PCM/擦り傷/外装焼け/12.1DSTN) | ¥19,800 |
| U50014 | PC-9821NR15/S10(Pen(150)/48MB/2.1GHD/CDドライブ欠品(外付けケーブル含む)/SVGA/外装焼け/ドット抜け有り/12.1TFT) | ¥27,800 |
| U6457 | PC-9821NR15/S14F(Pen(150)/48MB/1.4GHD/CD10/36.6K/SVGA/12.1TFT/32MB増設) | ¥42,800 |
| U6097 | PC-9821NW150/S20D(MP(150)/32MB/2GHD/CD16/SVGA/擦り傷/ロゴ部分塗装剥げ/12.1TFT) | ¥36,800 |

バッテリーパックは消耗品につき保証対象外とさせていただきます。

### MAC

| 注文No. | 商品 | 価格 |
|---|---|---|
| U9538 | PowerMac 7600/200(PPC604e(200)/128MB/2GHD/CD12/外装焼け/擦り傷) | ¥19,800 |
| U9458 | PowerMac 8500/132(PPC604(132)/80MB/1GHD/CD4/外装焼け/擦り傷/外装塗装剥げ) | ¥19,800 |
| U9527 | PowerMac 8500/150(PPC604(150)/160MB/1.2GHD/CD8/外装焼け/擦り傷) | ¥16,800 |
| U9540 | PowerMac 8500/180(PPC604e(180)/448MB/2GHD/CD8/擦り傷/ファイルカバー無し/ミドルタワー型) | ¥29,800 |
| U9380 | PowerMac 8600/250(PPC604e(250)/160MB/4GHD/CD24/外装焼け/擦り傷/ZIP付き(未チェックの為保証無し)) | ¥26,800 |
| U9339 | PowerMac 9500/200(PPC604e(200)/32MB/2GHD/CD8/外装焼け/擦り傷/汚れ) | ¥23,800 |
| U9461 | PowerMac 9600/300(PPC604e(300)/128MB/4GHD/CD24/外装焼け/ZIP付き(未チェックの為保証無し)) | ¥39,800 |
| U9528 | PowerMac G3/233(PPCG3(233)/160MB/4GHD/CD/外装焼け)……… | ¥36,800 |
| U1367 | PowerMac G3/DT266(PPCG3(266)/32MB/4GHD/CD24/ZIP付き(未チェックの為保証対象外)/外装キズ/外装焼け) | ¥39,800 |

キーボードは別売となります。

### PowerBook

| 注文No. | 商品 | 価格 |
|---|---|---|
| U9466 | PowerBook 1400CS/166(PPC603e(166)/48MB/1.3GHD/CD8/SVGA/擦り傷/外装焼け/11.3FTSN) | ¥29,800 |
| U3990 | PowerBook 1400C/117(PPC603e(117)/16MB/1GHD/CD6/SVGA/11.3TFT) | ¥34,800 |
| U6115 | PowerBook 1400C/117(PPC603e(117)/64MB/1GHD/CD6/SVGA/擦り傷/11.3TFT) | ¥37,800 |
| U3501 | PowerBook 5300Ce(PPC603e(117)/32MB/1.1GHD/擦り傷/外装ヤケ/10.4CTFTカラー) | ¥25,800 |
| U6950 | PowerBook G3 SERIES(G3-300)(PPCG3(300)/192MB/8GHD/CD/14.1TFT) | ¥89,000 |

### ディスプレイ

| 注文No. | 商品 | 価格 |
|---|---|---|
| U4929 | PC-9821CT(NEC)(17マルチ:1024×768max/外装焼け/外装カケ)……… | ¥7,900 |
| D0151 | JC-1736VM(NEC)(17マルチ:1024×768max/3列15pin)……… | ¥9,800 |
| U4468 | D1028L(DELL)(17マルチ:1024×768max/外装キズ/外装焼け/3列15pin)… | ¥7,900 |
| U4920 | 1400FP(DELL)(14TFT液晶:1024×768max/外装焼け/3列15pin)……… | ¥27,800 |
| U4891 | P991グレー(新古品未開封)◆お勧め(DELL)(19マルチ:1600×1200max/3列15pin/未開封)‥ | ¥39,800 |
| U4658 | PCXAV-YZ(DEC)(17マルチ:1280×1024max/外装焼け/3列15pin)……… | ¥7,900 |
| U4743 | DJ702E(MAG)(17マルチ:1280×1024max/外装焼け/擦り傷/3列15pin)…… | ¥8,900 |
| U4745 | L360(NANAO)(15TFT液晶:1024×768max/24～61kHz/外装焼け/キズ/3列15pin) | ¥29,800 |
| U4869 | FLEXSCAN E76D(NANAO)(21マルチ:1280×1024max/外装焼け/3列15pin) | ¥24,800 |
| U4547 | CPD-20SF3(SONY)(20マルチ:1280×1024max/外装焼け/3列15pin/5BNC) | ¥26,800 |
| U4253 | FMV-DP97Y1(富士通)(17マルチ:1280×1024max/外装焼け/外装キズ/3列15pin) | ¥10,800 |
| U4922 | FMV-DP9714(富士通)(17マルチ:1024×768max/外装焼け/擦り傷/3列15pin) | ¥7,900 |
| U4763 | RDT-151X(三菱)(15TFT:1024×768max/外装焼け/3列15pin)……… | ¥33,800 |
| U4350 | RD-21G(ダイヤモンド)(三菱)(21マルチ:1600×1200max/外装キズ/外装焼け/3列15pin) | ¥29,800 |
| U4930 | PC-DT3140(日立)(14.1TFT:1024×768max/外装焼け/バックパネル無/3列15pin) | ¥21,800 |
| U4877 | TX-D5L31A-J(パナソニック)(15TFT:1024×768max/外装焼け/3列15pin) | ¥27,800 |

### プリンタ

| 注文No. | 商品 | 価格 |
|---|---|---|
| U0462 | PC-PR201/87LA(NEC)(136桁シリアル:漢字87字/秒/外装焼け/カラー対応) | ¥39,800 |
| U0534 | PC-PR700X(NEC)(136桁シリアル:漢字80字/秒/外装焼け/水平プリンタ)‥ | ¥59,800 |
| U5180 | PR-D700XX(NEC)(ドットインパクトプリンター:A3/後オプションカバー無)…… | ¥49,800 |
| U5255 | PiCTY4000(PR-J4000)(NEC)(カラーインクジェット:～A3/1200×1200dpi/外装焼け:発売98/2) | ¥13,800 |
| U5144 | BJC-700J(キヤノン)(バブルジェットカラー:1200×600dpi/A4/重量6.4kg/外装焼け) | ¥16,800 |
| U5253 | LBP-470(キヤノン)(レーザーショット:A4対応/外装焼け)……… | ¥29,800 |
| P0142 | LBP-750(キヤノン)(レーザー:A3対応/付属品付)……… | ¥26,800 |

トナー等は消耗品につき標準添付されておりません。別途、ご確認下さい。

商品の状態により価格の変動がある事がございます。
詳しくはTELかネット上でご確認ください。

万一、品切れの際はご容赦ください。(3月28日現在)

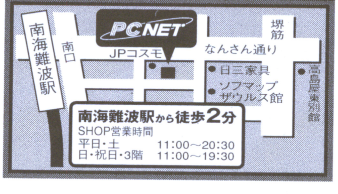
こちらの在庫リストは通販専用となります。店舗在庫は各店舗へご来店の上ご確認ください。

### 秋葉原駅前店

千代田区外神田1-15-6 ミツウロコビル4F
TEL.03-5209-6111

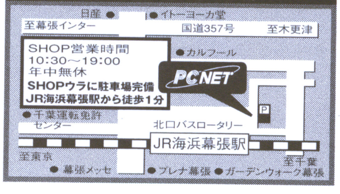
### 大阪日本橋店

浪速区難波中2-1-7 なんば東ビル
TEL.06-4396-1401




### 幕張店

千葉県千葉市美浜区ひび野1-9




## パシフィック ネット


株式会社パシフィックネット
本社 〒144-0051 東京都大田区西蒲田7-10-1
相互西蒲田ビル TEL.03-3733-1441 FAX.03-3733-1440
通販営業時間／月～金 9:00～19:00 土・日・祝日 11:00～19:30


計測機器を各ショップのみで販売中。
詳しくは各ショップまでお問い合わせください。

iモードでショップ最新情報検索
http://www.prins.co.jp/i/

## ◆ 定期購読●直接購読のしおり ◆

弊社刊行の雑誌は，ご自宅直送方式の年間定期購読をお勧め申し上げます。スピーディーで確実，そしてお得な定期購読をぜひご利用下さいませ。

また　ご希望の雑誌・書籍が，ご近所の書店にない場合は右の振替用紙にて直接ご注文下さい。

**■ 年間定期購読料（消費税込料金となっております。）**

| | | | |
|---|---|---|---|
| 電　気　店 | ￥16,000 | マイコンBASICマガジン | ￥6,120 |
| 電波新聞縮刷版（CD-ROM版） | ￥96,000 | | |

**■ 年間購読の特典**

①特別号も通常の定価でご購読いただけます。
②電気店は13冊分の購読ができます。

**◆確実な振替口座を利用してご送金下さい◆**

送金手数料が不要で，確実な郵便振替でご送金をお願い申しあげます。手続きは簡単です。右の郵便振替用紙に下記の必要事項を記入し，郵便局窓口に必要金額と共にご提出いただくだけですみます。

①購読申込書に必要事項を記入ください。なお1年間，或は2年間購読かをはっきりご記入ください。
②払込通知票，払込票に送金合計金額と住所，氏名をご記入ください。払込通知票と払込票は絶対に切りはなさないようにしてください。なお，必ずフリガナをご記入下さい。
③弊社への御意見・御希望がございましたら，通信欄へご記入下さい。
④すべて記入が終了しましたら，案内文のみ切り離して郵便局窓口に金額を添えて提出してください。

※振替によるご送金は，着本までに約2週間ほどかかりますのでお含みおきください。
**お急ぎの場合は，現金書留に，右購読申込書を同封してご送金ください。**

この受領証は、郵便局で機械処理をした場合は郵便振替の払込みの証拠となるものですから大切に保存してください。

ご注意

この払込書は、機械で処理しますので、口座番号及び金額を記入する際は、枠内に丁寧に記入してください。

また、下部の欄（表面及び裏面）を汚したり、本票を折り曲げたりしないでください。

（郵政事業庁）

# 払込取扱票

通常払込料金加入者負担

02 東京

口座番号（右詰めにご記入ください）

00150-3-51961

金額 千 百 十 万 千 百 十 円 ※

加入者名 株式会社 電波新聞社

料金　特殊取扱

各票の※印欄は、払込人において記載してください。

通信欄

| ※定期購読誌名 | | 購読期間 | 送金額（送料当社負担） |
|---|---|---|---|
| | *新規 継続 | 年　月号より　年間 | 円 |
| 御希望誌・ソフト・グッズ名 | | 冊数 | 送金額（送料共） |
| | | | 円 |
| | | | 円 |

| フリガナ 送付先住所 □□□-□□□□ | 産業分類 | 所属又は職種 |
|---|---|---|
| フリガナ | 電話 TEL | 年令 才 / 性別 男・女 |

払込人住所氏名（郵便番号　）※

（電話番号　－　－　）

受付局日附印

裏面の注意事項をお読みください。郵政事業庁（私製承認 東 第1005号）

切り取らないで郵便局にお出しください。

# 払込票兼受領証

口座番号 00150-3 右詰めにご記入ください 51961　通常払込料金加入者負担

加入者名 株式会社 電波新聞社

金額 千 百 十 万 千 百 十 円 ※

払込人住所氏名 ※

料金（消費税込み）円

特殊取扱

受付局日附印

記載事項を訂正した場合は、その箇所に訂正印を押してください。

切り取って郵便局へお出し下さい。

# 電波新聞社出版物一覧

（消費税込料金となっております）

2002. 3 現

| | | | | | |
|---|---|---|---|---|---|
| 最新エレクトロニクス用語事典 | 5,300円 | 〒380円 | F-BASIC V6.3プログラミングのすべて | 2,835円 | 〒310 |
| 半導体・ICのすべて | 2,520円 | 〒310円 | まんがばそてんシリーズ | 各 | 各 |
| ビデオ技術のすべて | 2,100円 | 〒310円 | 第1巻(基礎編)/第2巻(応用編)/第3巻(ネットワーク編) | 1,260円 | 〒310 |
| TVのすべて改訂版 | 1,988円 | 〒310円 | パソコン認定試験合格対策試験問題集3・4級 | 2,100円 | 〒310 |
| DVD&ホームシアター | 1,680円 | 〒310円 | シスアド受験スーパー記憶術 | 2,900円 | 〒340 |
| 新デジタル映像技術のすべて | 2,100円 | 〒310円 | 基本情報受験スーパー記憶術 | 2,835円 | 〒340 |
| センサーのすべて | 2,000円 | 〒310円 | 初級シスアド受験マニュアル | 3,780円 | 〒450 |
| 改訂版DVD&DVC入門基本18章 | 1,529円 | 〒310円 | 基本情報受験マニュアル | 3,780円 | 〒450 |
| 増補改訂版デジタル放送がわかる本 | 1,600円 | 〒310円 | デジタル1種〈技術科目〉最新テキスト | 2,205円 | 〒340 |
| 増補改訂版・新ビデオ技術ハンドブック | 2,500円 | 〒310円 | デジタル3種受験マニュアル | 2,940円 | 〒340 |
| 新・まんが電気学入門 | 1,500円 | 〒310円 | アナログ・デジタル全資格〈法規科目〉 | 1,995円 | 〒340 |
| 新電気学入門講座 | 2,050円 | 〒310円 | 特殊無線技士受験マニュアル・2002年版 | 2,940円 | 〒340 |
| 新オシロ・ロジアナ入門講座 | 2,243円 | 〒310円 | オールアバウト カプコン対戦格闘ゲーム1987～2000 | 2,100円 | 〒450 |
| 電子計測器ガイドブック | 7,500円 | 〒450円 | オールアバウト SNK対戦格闘ゲーム 1991～2000 | 2,310円 | 〒450 |
| よくわかるプラズマテレビ(PDP) | 1,680円 | 〒310円 | 電子パーツ物知り百科 | 1,890円 | 〒310 |

※上記以外にも新聞・出版物がございますので最寄りの本社、支局へお問い合わせ下さい。

| | | | | | |
|---|---|---|---|---|---|
| 東京本社 | 03-3445-8201(ダイヤルイン) | 新潟支局 | 025-245-2526 | 京都支局 | 075-221-8021 |
| 大阪本社 | 06-6203-3361 | 関東総局 | 027-326-3206 | 広島支局 | 082-228-558 |
| 西部本社 | 092-431-7411 | 松本支局 | 0263-36-2266 | 高松支局 | 087-861-3111 |
| 札幌支局 | 011-641-5591 | 名古屋支局 | 052-261-4541 | 鹿児島支局 | 099-226-3630 |
| 仙台支局 | 022-227-7211 | 金沢支局 | 076-263-8661 | | |

キリトリ線

# 月刊マイコンBASICマガジン⑤ 愛読者カード

★OFコーナー宛 ♥あくせすROOM宛 ■質問・感想(本誌への掲載／可・不可)

ペンネーム：

## ■アンケート回答記入欄 ■ ※黒の筆記用具でわかりやすく，はっきりとご記入ください

| | | | |
|---|---|---|---|
| Q1 [ ] | Q2 [ ] ［メーカー名 ］ | | |
| Q3 [ ] | Q4 [ ] | Q5 [ ] | |
| Q6 [ ][ ][ ][ ][ ][ ][ ][ ] | | | |
| Q7 [ ][ ][ ][ ][ ] ［商品名 ］ | | | |
| Q8 [ ][ ][ ][ ][ ][ ][ ][ ] | | | |
| Q9 [ ][ ][ ][ ][ ][ ][ ][ ] | | | |
| Q10 [ ] | Q11 [ ] ［④の人はソフト名を記入 ］ | | |
| Q12 ［ ］ | Q13 ［ ］ | | |
| Q14 [ ][ ][ ] | Q15 [ ][ ][ ] | | |

## ■アンケートにご協力ください。

※左の愛読者カード・記入欄に，以下の質問に該当する番号をご記入ください。複数回答の質問で，回答欄をすべて埋める必要はありません。

Q1　メインで使っているパソコンは？
①デスクトップ ②ノート ③持っていない

Q2　パソコンの種類は？（Q1で③以外の方）
①自作 ②ショップブランド ③メーカー製
※枠内にメーカー名も記入してください

Q3　メインで使っているOSは？（Q1で③以外の方）
①Windows 95 ②Windows 98 ③Windows Me ④Windows 2000 ⑤Windows NT ⑥Windows XP ⑦Macintosh ⑧UNIX ⑨Linux ⑩その他

Q4　パソコン歴は？
①1年未満 ②1～2年 ③3～5年 ④6～10年 ⑤11年以上 ⑥なし

Q5　インターネット歴は？
①1年未満 ②1～2年 ③3～5年 ④6～10年 ⑤11年以上 ⑥なし

Q6　パソコンの主な使用目的は？（複数回答可）
①プログラミング ②ゲーム ③インターネット ④メール ⑤ミュージック ⑥学習 ⑦CG ⑧その他

Q7　これから買いたいと思うコンピュータ機器は？（複数回答可）
①パソコン本体 ②プリンタ ③スキャナ ④デジタルカメラ ⑤その他
※わかれば，商品名も記入してください

Q8　持っているゲーム機は？（複数回答可）
①プレイステーション（PSone含） ②プレイステーション2 ③ドリームキャスト ④ニンテンドウ64 ⑤ゲームキューブ ⑥Xbox ⑦ゲームボーイ ⑧ゲームボーイカラー ⑨ゲームボーイアドバンス ⑩ワンダースワン ⑪ワンダースワンカラー ⑫その他

Q9　これから買いたいと思うゲーム機は？（複数回答可）
①プレイステーション（PSone含） ②プレイステーション2 ③ドリームキャスト ④ニンテンドウ64 ⑤ゲームキューブ ⑥Xbox ⑦ゲームボーイカラー ⑧ワンダースワンカラー ⑨ゲームボーイアドバンス

※裏面へつづく→

キリトリ線

## ■メモ欄…プログラム担当者・編集部へのご意見・ご感想など

Q10　携帯・PHSを持っていますか？
①両方持っている ②携帯 ③PHS ④購入予定 ⑤予定なし

Q11　ワープロ・ソフトは何をお使いですか？
①Word ②一太郎 ③メモ帳 ④その他

Q12　使用中のプログラミング言語は？
※枠内に一つご記入ください

Q13　これから勉強したいプログラミング言語は？
※枠内に一つご記入ください

Q14　今月号の本誌で面白かった記事を三つ，下の表からお答えください

Q15　今月号の本誌でつまらなかった記事を三つ，下の表からお答えください

ご協力ありがとうございました。

## ■今月号の記事一覧


① 編集部の学校訪問記…6
② 一歩先ゆくホームページ作り…10
③ CPUの歴史と未来を探る！…21
④ ベーシックプログラミング教室…29
⑤ はじめようMicrosoft OfficeでBASICプログラミング…35
⑥ ＰＣハードウエアテクニック…39
⑦ 特選投稿プログラム・コーナー…45
⑧ 明日のスター・プログラマー…97
⑨ Dr.Dのベスト・プログラム研究室…98
⑩ マンガで学ぶ! コンピュータの世界!…100
⑪ これからはじめる ブロードバンド活用法…104
⑫ チャレンジ! オリジナル プログラム言語の制作…109
⑬ HSPで楽々 ３Dプログラミング！…114
⑭ Visual Basicアルゴリズム講座…117
⑮ 21世紀の遊びはこれロボ道場…122
⑯ .NETプログラミング入門…126
⑰ 電子工作にチャレンジ! 自作一代記…131
⑱ あくせすROOM/プレゼント…134
⑲ SOFTWARE MAKER HOT INFORMATION…135
⑳ 表紙
㉑ OVER FLOW


## マイコンBASIC Magazine　プログラム投稿応募カード

| | | | | | |
|---|---|---|---|---|---|
| 言語名（音源名） | | | | | |
| プログラム名（曲名） | | | | | |
| 氏　名 | | ペンネーム | | 年齢 | 歳 |
| 住　所 | 〒 | | | | |
| TEL | ※連絡可能な時間帯 | | 応募年月日 | 年　月　日 | |

キリトリ線

郵便はがき

50円切手をお貼り下さい

141-8715

（受取人）

東京都品川区東五反田1－11－15

電波新聞社

月刊 マイコンBASICマガジン 編集部

5月号 行

| | | | |
|---|---|---|---|
| (フリガナ) | | 年齢 | 性別 |
| ご氏名 | | 歳 | 男・女 |
| □□□-□□□□ | | 電話 | |
| ご住所 | 都・道 府・県 | | |

職業：小学生／中学生／高校生／高専生／短大生／大学生／専門学校生／予備校生／アルバイト／公務員／会社員／その他（　　　）

本誌を 毎月・時々・初めて 読む　　何人で読みますか（　）人

自分のホームページを　持っている・持つ予定・持っていない・予定なし

本誌のほかによく読む雑誌は？［　　　　］

| | | |
|---|---|---|
| ベーマガお楽しみプレゼント⑤ 応募欄（134ページ掲載） | 希望番号 | 品名 |
| | 番 | |

# パソコンRPG入門最適！

PCゲームの定番王「イースⅠ完全版」オマケ付
新価格 3,800円！（ファルコム通販価格）

## 新価格WindowsXP対応 好評発売中！

# イースⅠ完全版

初回限定 オマケ付

DVD-ROM版／CD-ROM版 ¥3,800（ファルコム通販価格）

※新価格XP対応「イースⅠ完全版」は、「イースⅠ&Ⅱ完全版」の「イースⅠ」と同じものです。ゲーム内容に変更はありません。

スピーディーに展開するドラマティックな物語。独自の世界観を構築するグラフィックと音楽。他に類をみない抜群の操作性と爽快感。RPGブームの先駆けであり、全てのストーリー型RPGの基礎となったゲーム、それが「イース」です。イージーからハードまで難易度が選択できるから、RPG初心者でも必ずエンディングまでプレイできます。多くのユーザーが魅了され続け、いまだ語り継がれる“伝説”を、ぜひあなたも“経験”してください。

### 【オマケ❶】ファルキャラトレーディングコースター

トレーディングカードのように集めて楽しい新感覚超極薄コースター（90×90mm/厚さ0.5mm）。紙製ではないので汚れてもサッと一拭きで繰り返し使えます。裏面は貼り跡の残らない特殊粘着加工。フィルムをはがせばデスクやテーブルにピタッと貼りつき安定感バツグン。カードのように薄いので額に入れて飾るなどお部屋のインテリアにも最適。（別売:1パック3枚入480円）

### 【オマケ❷】イースたためるCDスタンド

イースエンブレムがお洒落な最大9枚収納できる折りたたみ式CDスタンド。蓋をたたむと普通のCDプラケースと同じサイズになり、使わないときも邪魔にならないスグレもの。（別売:1,200円）

### 【オマケ❸】イースチビデカ2wayマット

チビマウスマットにも、デカコースターにもなる超便利マット（120×120mm）。クッション付で滑らない安心の使い心地。この便利さ使った人しかわからないかも。（別売:480円）

※オマケ1～3は別売りもあります。ご購入はファルコム通販をご利用ください。

初回限定
★オマケ★

http://www.falcom.co.jp/

七瀬葵 描き下ろし 新イラスト NEWパッケージで登場！

## PCゲーム売上ランキング1位！

週刊アスキー「今週の売れ筋ソフト」（2001年12/17～12/30集計）、CD-ROM Fan「売れ筋ランキング」（2001年12月集計）
TECH Win「SALES RANKING」（2001年12/1～12/31集計）、電撃王「PCゲーム売り上げチャートBEST10」（2001年12/1～12/31集計）

新価格版 好評発売中

攻略ガイドも発売中。完全制覇を目指せ！

Windows 95/98/Me/2000/XP
DVD-ROM版 CD-ROM版

■お求めは取扱い店またはカンタン便利なファルコム通販（falcom.comファルコムドットコム）もご利用下さい。■社員募集（新卒・既卒・中途）！詳しくはホームページ「会社案内・社員募集」をご覧ください。

Falcom®
NIHON FALCOM CORPORATION
〒190-0012 東京都立川市曙町1-14-13
☎042(527)6501 http://www.falcom.com/

本能はワイルドに、感情はディープに。AIBO ERS-220起動。

商品およびご購入に関するお問い合わせ

□AIBOインフォデスク 0570-008844 [ナビダイヤル:全国どこからでも市内通話料でご利用いただけます] □携帯電話・PHSから 03-5470-0660 [受付時間:午前10時～午後6時(年末年始を除く)]

□インターネット http://www.aibo.com/ [ネットワークやサーバーなどのメンテナンス期間を除き24時間受付]

エンターテインメントロボット"AIBO"ERS-220 希望小売価格 180,000円(税別)

□AIBO本体のみでは基本的な動作しかできません。より個性的な動きをAIBOに組み込むためには別売ソフトが必要です。□商品の色は実際とは若干異なる場合があります。□"AIBO"および"AIBOロゴ"は、ソニー(株)の商標または登録商標です。※背景とAIBOは合成によるものです。実際の商品の大きさとは異なりますのでご注意ください。また、屋外では使用しないで下さい。製品に傷がついたり不具合の原因となることがあります。

aibo



第二十一巻第五号通巻二三九号　昭和五十七年十二月六日第三種郵便物認可　二〇〇二年四月二十五日印刷　二〇〇二年五月一日発行

電波新聞社 〒141-8715 東京都品川区東五反田一―十一―十五　電話〇三(三四四五)六一一一

定価550円 本体524円 雑誌18361-05 T1118361050554

2002

わかる！動かせる！
プログラムが組める雑誌

# マイコンBASIC Magazine

5

大好評連載！

## 一歩先ゆくホームページ作り②

ルビ／段組／スクロール

文字・文章―いろいろな書き方・見せ方

http://www.basicmagazine.net/

電波新聞社